Finance and Economics Discussion Series

Federal Reserve Board, Washington, D.C.
ISSN 1936-2854 (Print)
ISSN 2767-3898 (Online)

Linear and nonlinear econometric models against machine learning models: realized volatility prediction

Rehim Kilic

2025-061

Please cite this paper as:
Kilic, Rehim (2025). "Linear and nonlinear econometric models against machine learning models: realized volatility prediction," Finance and Economics Discussion Series 2025-061. Washington: Board of Governors of the Federal Reserve System, https://doi.org/10.17016/FEDS.2025.061.

Linear and nonlinear econometric models against machine learning models: realized volatility prediction

Rehim Kılıç*

July 2025

Abstract

This paper fills an important gap in the volatility forecasting literature by comparing a broad suite of machine learning (ML) methods with both linear and nonlinear econometric models using high-frequency realized volatility (RV) data for the S&P 500. We evaluate ARFIMA, HAR, regime-switching HAR models (THAR, STHAR, MSHAR), and ML methods including Extreme Gradient Boosting, deep feed-forward neural networks, and recurrent networks (BRNN, LSTM, LSTM-A, GRU). Using rolling forecasts from 2006 onward, we find that regime-switching models—particularly THAR and STHAR—consistently outperform ML and linear models, especially when predictors are limited. These models also deliver more accurate risk forecasts and higher realized utility. While ML models capture some nonlinear patterns, they offer no consistent advantage over simpler, interpretable alternatives. Our findings highlight the importance of modeling regime changes through transparent econometric tools, especially in real-world applications where predictor availability is sparse and model interpretability is critical for risk management and portfolio allocation.

JEL Classification: C10, C50, G11, G15.

Keywords: Realized volatility, machine learning, regime-switching, nonlinearity, VaR, forecasting.

*Federal Reserve Board, Washington, DC E-mail: rehim.kilic@frb.gov. The views presented in this paper are solely those of the author and do not represent those of the Board of Governors or any entities connected to the Federal Reserve System.

1 Introduction

Accurate volatility forecasting is vital for assessing systemic risk, asset pricing, portfolio allocation, and risk management. Since the foundational work of Engle (1982), Bollerslev (1986), and Taylor (1982), volatility modeling has evolved significantly but remains a challenging task, as discussed in Bauwens et al. (2012) and Takahashi et al. (2023).

Recently, machine learning (ML) methods have gained traction in financial econometrics as tools for modeling complex, nonlinear dynamics in volatility—particularly during episodes of clustering or regime shifts. Empirical studies have explored whether ML models can outperform traditional approaches like the heterogeneous autoregressive (HAR) model (Corsi , 2009).[1] However, the evidence remains mixed: while ML models can capture flexible interactions and high-dimensional predictors, they do not always yield consistent gains over linear benchmarks.

The recent literature on realized volatility (RV) forecasting reveals substantial heterogeneity over modeling approaches, data environments, and findings (see, Section 2). A wide array of ML models has been explored, including regularized regressions (e.g., LASSO, Elastic Net), tree-based methods (Random Forests, Gradient Boosted Trees), and neural networks—both feed-forward and recurrent, such as LSTM and NARX architectures. Studies also vary considerably in their use of data, spanning low-frequency monthly RV series with rich macroeconomic and sentiment predictors (Mittnik et al., 2015; Bucci , 2020), to high-frequency RV measures at daily or intraday intervals (Christensen et al., 2023; Rahimika and Poon , 2024). Forecasting targets range from firm-level RV to aggregate index volatility, and from short (1-day) to longer (monthly) horizons. Despite this diversity, the comparative performance of ML models relative to traditional econometric approaches—particularly the linear HAR model—remains mixed. While some studies report modest gains from ML models when additional predictors or cross-sectional information are leveraged (Christensen et al., 2023; Zhang etal. , 2023), others find that well-specified and frequently re-estimated HAR-lineage of models remain difficult to beat (Audrino etal., 2020; Branco et al., 2024).

Notably, aside from Bucci (2020)—who, unlike our study, relies on monthly data—recent studies have not systematically compared ML models to nonlinear econometric models, particularly those incorporating regime-switching dynamics such as threshold or smooth-transition HAR variants. This represents a critical gap in the literature, as many ML methods are designed to capture structural nonlinearities that are also explicitly modeled by such econometric frameworks. Our study addresses this gap by being the first to evaluate the relative performance of a broad suite of ML models against both linear and nonlinear econometric models—including regime-switching HAR variants—using high-frequency RV data. In addition to established linear models such as ARFIMA (Baillie , 1996; Andersen et al., 2001) and HAR (Corsi , 2009),

[1] For an overview of these developments, see Section 2 and Gunnarsson et al. (2024) for a detailed review of this growing literature.

we examine three nonlinear HAR extensions: the Markov-switching HAR (MSHAR), threshold HAR (THAR), and smooth-transition HAR (STHAR), all of which are designed to capture potential regime-dependent dynamics in volatility. For ML models, we include tree-based methods such as Extreme Gradient Boosting (XGB) (Hastie et al., 2009), deep feed-forward neural networks (DNNs), and several recurrent neural network (RNN) architectures—basic RNN (BRNN), gated recurrent unit (GRU) (Chung et al., 2014), long short-term memory (LSTM), and LSTM with attention (LSTM-A) (Goodfellow et al., 2016; Lipton et al., 2015). Importantly, our empirical framework spans the period from 1996 through recent years, enabling us to assess model performance across distinct states of the economy and financial markets—including the Global Financial Crisis (GFC), the COVID-19 shock, and the post-pandemic normalization and monetary tightening. This allows for a robust and systematic comparison of models under varying volatility regimes, structural dynamics, and predictor environments.

By juxtaposing relatively complex ML algorithms with intuitive nonlinear econometric models, this study offers fresh perspectives on whether the additional complexity of ML methods yields meaningful forecasting improvements—particularly compared to simpler and intutive nonlinear econometric alternatives, an area that remains under explored in the literature. The findings aim to assist both researchers and practitioners in selecting appropriate forecasting tools for realized volatility, where the trade-offs between model interpretability, accuracy, and complexity are crucial.

We adopt a yearly rolling training and testing approach, beginning with an initial period from 1996 to 2005 and testing through various market conditions, including the Global Financial Crisis (GFC), the COVID-19 shock, and the recent interest rate hike. This framework allows for comprehensive model evaluation over different volatility regimes. We assess performance across three dimensions: statistical accuracy, Value at Risk (VaR) forecasting, and realized economic utility. We use mean squared prediction error (MSPE) and quasi-likelihood (QLIKE) of Patton (2011), and the model confidence set (MCS) procedure of Hansen et al. (2011). Diebold and Marioano (1995) (DM) tests are employed for equal predictive ability testing, and VaR accuracy is evaluated using the DM and coverage tests of Kupiec (1995) and Christoffersen (1998).

Our analysis yields six key findings. First, under the baseline scenario—where only past values of realized volatility (RV) are used as predictors—nonlinear regime-switching models, particularly THAR and STHAR, consistently outperform both the linear HAR model and a broad class of ML models, across both tranquil and volatile periods such as the Global Financial Crisis and the COVID-19 shock. Second, while ML models are capable of capturing certain nonlinear patterns in the data, they do not demonstrate consistent superiority over econometric models in forecasting RV. Third, STHAR also performs best in Value-at-Risk (VaR) forecasting, achieving statistically accurate coverage across nearly all periods, both under

the baseline and when augmented with additional macro-financial predictors. Fourth, although the inclusion of additional predictors narrows the gap in predictive accuracy across models, ML approaches still fail to consistently outperform nonlinear econometric models—especially THAR and STHAR. Fifth, STHAR yields the highest realized utility under the baseline scenario, with performance differences across models diminishing when the predictor set is expanded. Finally, taken together, our results indicate that ML models are not a one-size-fits-all solution to the challenges of RV prediction. They reinforce recent evidence in the literature (Branco et al., 2024) showing the competitiveness of linear HAR-type models and further demonstrate the strong performance of simple, interpretable nonlinear regime-switching models. Notably, our results show that regime-switching models outperform a broad set of ML approaches ranging from XGB to DNNs as well as several RNNs, particularly during periods of extreme market conditions—when RV spikes or settles into calm regimes—as they more effectively capture shifts in underlying volatility dynamics. These findings underscore the practical value of explicitly modeling structural shifts in volatility through econometric specifications, as opposed to relying solely on flexible but opaque ML architectures. In particular, when the predictor set is sparse—a common setting in real-world risk management applications—nonlinear econometric models such as THAR and STHAR offer a robust and effective alternative to more complex machine learning methods.

The remainder of the paper is organized as follows. Section 2 presents a short overview of the recent literature on volatility prediction using machine learning. Section 3 describes the dataset. Sections 4 and 5 introduce the linear, nonlinear econometric, and ML models, respectively. The main empirical results under the baseline scenario—where predictors include only the past history of realized volatility—are presented in Section 6. Section 7 concludes. Details of our estimation, training, cross-validation, and hyper-parameter tuning approaches, as well as performance evaluation metrics, are discussed in Appendix A. The robustness checks, including results with additional predictors and an extended analysis of recurrent neural networks with longer time steps for realized volatility, are provided in Appendices B and C, respectively.

2 Related Literature

This paper is closely related to the literature on forecasting aggregate stock index volatility and specifically the recent papers that explore capability of various ML algorithms in forecasting RV. The literature on volatility modeling and prediction is large and goes back to Engle (1982), Bollerslev (1986), and Taylor (1982), who developed GARCH and stochastic volatility models for conditional volatility which provide forecasts of daily volatility from daily return. As high-frequency data has become available, measures based on quadratic variation, such as RV has heavily been used in modeling and predicting market volatility. Numerous models have been

proposed over the years (Bauwens et al., 2012; Takahashi et al., 2023). With the rise of applications of ML techniques, an increasing number of papers in the recent years, has been offering new set of tools in the context of volatility prediction. Gunnarsson et al. (2024) provide an excellent survey of this recent and growing literature. Our work relates and contributes to this recent line of research that investigates the performances of a number of econometric models and ML models in predicting market volatility and specifically RV that is constructed using high-frequency data on aggregate market index returns. In the following we summarize key results from this recent literature.

We use two benchmark econometric models, namely ARFIMA and HAR models. Andersen et al. (2003) showed that ARFIMA model outperforms GARCH and related models in predicting daily RVs. HAR model as proposed by Corsi (2009) is a parsimonious alternative to ARFIMA which can capture the long-memory and perform well-in forecasting daily RV. Given the relatively simpler framework the HAR model is based on and its' success in modeling and forecasting RV, it has become a "benchmark" (see, for example Hansen and Lunde , 2005). Several extensions of HAR model have also been introduced including extensions to capture leverage effects, (see, for example Corsi and Renó , 2012; Patton and Sheppard , 2015) among others. More recently Izzeldin et al. (2019) showed that ARFIMA and HAR models models perform equally well in predicting RV. In this paper, we use the original HAR model and a simple ARFIMA specification when features set includes only the past values of RV and consider extensions of HAR and ARFIMA models when set of features include additional predictors measuring financial and macroeconomic conditions.

The recent literature primarily compares the performance of linear econometric models against ML models, especially when the predictor set includes features beyond past values of RV. Some studies use low-frequency monthly data to increase the number of predictors, despite a few recent exceptions leveraging daily or higher frequency data. Hamid and Iqbal (2004) is one of the early works that studies the forecasting performance of neural networks using implied volatility and realized volatility of S&P 500 index prices at daily frequency, showing that neural networks outperform implied volatility forecasts and perform on par with RV.

Fernandes etal. (2014) extends the HAR model with additional predictors in the context of a neural network. Mittnik et al. (2015) demonstrates that a Random Forest (RF) approach with several macroeconomic predictors, such as VIX and TED spread, outperforms GARCH-family models at monthly horizons ranging from one to six months. Similarly, Audrino and Knaus (2016) applies the least absolute shrinkage and selection operator (LASSO) for RV prediction and shows that although HAR and LASSO suggest different lag structures, both perform similarly in terms of out-of-sample forecasting accuracy.

Audrino etal. (2020) extends LASSO by including additional features derived from economic variables and investor attention and sentiment measures, showing improved forecast accuracy

compared to the HAR model. Liu etal. (2018) report that simple RNNs can improve forecast accuracy of RV when the training sample is small, while the HAR model outperforms RNNs when the training sample increases. Bucci (2020) investigates the performance of linear, ARFIMA, HAR, and Logistic Smooth Transition Autoregressive (LSTAR) models against several neural network models (both feed-forward and recurrent), including long short-term memory (LSTM) networks and nonlinear autoregressive models with exogenous input (NARX) networks to forecast the monthly RV (estimated using daily returns) of the S&P 500 index. His results demonstrate that recurrent neural networks (RNNs), including LSTM and NARX, outperform econometric models in forecasting monthly volatility of the S&P 500 index.

Christensen et al. (2023) report significant gains using several feed-forward neural network models in predicting RV of the Dow Jones Industrial Average Index constituents. They compare ML models, including regularization approaches, regression trees, and feed-forward neural networks, to the HAR lineage of models and report that ML models outperform the HAR lineage, both when predictors are daily, weekly, and monthly lags of the RV, and when additional firm-level, macroeconomic, and financial predictors are included, especially over longer horizons.

Zhang etal. (2023) compare the HAR model with a wide range of ML models, including LASSO, Elastic Net, Partial Least Squares, random forests (RFs), stochastic gradient boosting (SGBs), and feed-forward deep neural networks (DNNs). They employ both single time series forecasting and panel data forecasting using the same ML methods to improve short-term forecasting accuracy, finding that panel-data-based ML methods outperform other models. Zhang etal. (2023) on the other hand, show that NNs dominate linear regressions and tree-based models in their ability to forecast intraday RV.

Rahimika and Poon (2024) compares the performances of the HAR and LSTM models for forecasting RV of 23 NASDAQ stocks using a very large predictor set, including variables extracted from limit order books and news, and reports superior performance of the LSTM model, especially when actual volatility is not extreme. By focusing on the fitting scheme in terms of training window and re-estimation frequency for the HAR model, Audrino etal. (2020) compares the forecasting performance of the HAR model against ML techniques, including Lasso, Random Forest (RF), Gradient Boosted Trees (GBT), and feed-forward NNs, using high-frequency data for 1,445 stocks between 2015 and 2023. They report that despite extensive hyperparameter tuning, ML models fail to surpass the linear benchmark set by the HAR model when a refined fitting approach is used for HAR. Their findings highlight the importance of fitting scheme, particularly re-estimation frequency and training window size, in driving the performance of the HAR model against ML algorithms.

Branco et al. (2024) provide a recent and methodologically relevant contribution. They compare linear HAR and HARX models against various ML models—including neural networks

and tree-based ensembles—for ten global stock indices. Their findings suggest that nonlinear ML models do not statistically outperform linear HAR variants, especially when additional predictors are used. However, their analysis is limited to linear HARX-type baselines and fixed NN architectures.

Relative to this literature, our paper offers six key advances. First, we benchmark ML models not only against linear HAR and ARFIMA but also against nonlinear HAR variants (THAR, MSHAR, STHAR), an area not explored in recent studies including Branco et al. (2024). The only study that explore nonlinear regime-switching models is Bucci (2020), which however, uses monthly data and focuses on a smooth transition model only.

Second, while most studies—including Christensen et al. (2023), Branco et al. (2024), and Rahimika and Poon (2024)—use fixed ML architectures, we dynamically select network structures over time. We train, validate, and test our models annually since 2006, allowing for performance tracking over distinct market conditions. This approach requires retraining and validation over time but enables a more dynamic comparison across different market environments, and hence assessment of performances of ML and econometric models in a comprehensive manner.

Third, the set of ML models we consider in this paper, includes an example from tree-based model and feed-forward neural networks as well as four RNNs that are primarily designed for sequential data with persistence. In this sense, our work is closely related to Audrino etal. (2020) and Rahimika and Poon (2024) as these papers also include RNNs. Differently from these papers, in addition to LSTM, we use three additional RNNs including LSTM with an attention mechanism, a basic RNN, and a simplified LSTM, called gated RRN, (GRU). Fourth, we focus on high-frequency RV forecasting for a single aggregate index—the S&P 500—which enables a more detailed analysis of within-series dynamics and facilitates sharper inference over an extended period that spans a wide range of volatility regimes.

Fifth, in terms of findings, our results suggest that relatively simple econometric models incorporating regime-switching dynamics can outperform machine learning models and more complex RNNs across several performance measures. While Branco et al. (2024) closely aligns with our study, our results emphasize the superior performance of nonlinear threshold and smooth transition HAR models, which consistently outperform both linear and nonlinear ML models when using only HAR variables as predictors. Even with an extended feature set, nonlinear econometric models such as THAR and STHAR remain highly competitive. These results highlight the critical importance of benchmarking ML models against nonlinear econometric alternatives to accurately assess the benefits of ML algorithms in realized volatility (RV) prediction and market risk measurement. Our findings suggest that, especially when compared to relatively simple and intuitive regime-switching nonlinear models, the incremental gains from

complex ML approaches may not justify their use in practical applications of RV prediction and risk management.

Finally, in most practical applications of RV modeling and forecasting, market practitioners often lack access to long time series of auxiliary predictors and must rely primarily on a few key risk factors. This constraint underscores the relevance of models that use sparse predictor sets and highlights the importance of our baseline findings—where ML models offer limited performance gains—relative to studies that rely on large sets of predictors to demonstrate marginal improvements.

3 Data: Realized Volatility and Predictors

Our objective is to explore the performances of linear and nonlinear econometric and machine learning (ML) models in predicting the daily realized variance, RV_t which can be thought to be an estimator for the integrated variance or the quadratic variation (IV_t) as defined in

$$IV_t = \int_{t-1}^{t} \sigma_s^2 ds, \tag{1}$$

where t is the day, σ_s is the instantaneous stochastic volatility (see, Andersen et al., 2001; Barndorff et al. , 2002, for the details of the definition and underlying price process for an asset). An estimator of IV is given by the realized variance:

$$RV_t = \sum_{s=1}^{n} r_s^2 \tag{2}$$

where n is the number of intraday logarithmic returns, and $r_s = \log\left(\frac{P_s}{P_{s-1}}\right)$.

In this paper, the realized volatility measure used is based on five-minute logarithmic price return with P_s and P_{s-1} denoting the first and the last price within the five-minute interval.[2] We construct daily RV estimates by using 5-minute intraday data on open and close prices of S&P 500 index which are obtained from Bloomberg. The sample period for the RV data covers the period from January 1996 to December 2023, and therefore include several periods of increased volatility and market stress, including the Global Financial Crisis (2007–2008), the European sovereign debt crisis (from 2009 to late 2010s, peaking in 2012), the Chinese stock market bubble (2015–2016), Brexit (2016–2020), the COVID-19 pandemic from 2020, and the monetary policy tightening period since early 2022.

[2] Andersen et al. (2001) and Barndorff et al. (2002) showed that the above sum of squared returns is a consistent estimator of the unobserved IV. Using 5-minute sub-sampling frequency is widely accepted time interval in the literature. For example, Liu et al. (2015) show that 5-minute sub-sampling frequency significantly outperforms other sub-sampling intervals in forecasting daily RVs.

In terms of set of predictors, we conduct our evaluation of models under two scenarios, one that only uses past values of RV and another where we include additional predictors. Under the first scenario, our baseline for the purpose of model comparison, we include the three moving averages of past RV used in the HAR model of Corsi (2009) and its nonlinear extensions, which correspond to horizons of 1, 5, and 22 days. The auto-regressive fractionally integrated moving average (ARFIMA) model of Andersen et al. (2003) essentially uses the past values of RV. In the threshold and smooth transition nonlinear extensions of HAR model (i.e., referred in the paper by THAR and STHAR, respectively), we use $d-$ day lagged relative RV for $d = 1, \cdots, 5$ as candidates for the threshold/transition variable. This variable essentially captures episodes of deceleration and acceleration of market volatility depending on the size of the deviation of past relative change in RV from the estimated threshold values in the context of THAR and STHAR models.[3]

Under this scenario, all models are estimated and trained by using past values of RV as there are no additional information used apart from the past history of RV itself. This baseline scenario is useful and important for the purpose of assessing the relative performance of linear and nonlinear econometric and ML models in predicting RV as some uses of RV, such as estimating market risk by using historical simulation Value at Risk (VaR) approach by market participants typically rely only on the past historical information on risk factors such as equity price returns.

To asses the robustness of our key findings under the baseline scenario, we consider a second scenario, motivated by the empirical literature (see, for insatnce, the recent studies Bucci , 2020; Christensen et al. , 2023; Zhang etal. , 2023; Branco et al. , 2024; Rahimika and Poon , 2024). Under this scenario we use a relatively large set of variables as potential predictors. Details of the extended set of predictors with our main findings are reported and discussed in Appendix B for the sake of brevity.

4 Econometric models

Given our objective of understanding the performances of pure time series econometric models and ML models in predicting RV, we consider econometric models that are specifically capable of characterizing temporal dynamics, long-range-dependence and possible extensions of such models to capture potential nonlinear dynamics. In this section, we provide an overview of these econometric models and refer readers to the key references for details.

[3]We explored using $d-$lagged RV itself as a transition or threshold variable but found that nonlinear least squares (NLS) algorithm converges consistently both THAR and STHAR models under the $d-$day lagged RV over each model training/estimation period.

4.1 Linear models

4.1.1 Auto-regressive Fractionally Integrated Moving Average Model

To this end, the first model we use is the Autoregressive Moving Average Fractionally Integrated, $ARFIMA(1, d, 1)$, model which allows for joint modeling of short term dynamics via autoregressive (AR) and moving average (MA) terms while capturing long-memory with the fractional integration parameter d (see, Andersen et al. , 2003).[4] The $ARFIMA(1, d, 1)$ model is given by

$$(1 - \phi L)(1 - L)^d RV_t = (1 + \theta L)u_t \tag{3}$$

where $1-\phi L)$ is the first order autoregressive polynomial in the lag operator L, $1+\theta L)$ is the first order moving-average lag polynomial, $-0.5 < d < 0.5$ is the fractional integration parameter, and u_{t+1} is the error term. Baillie (1996) provides an excellent discussion of the model, its' properties, autorrelation functions, estimation Maximum Likelihood Estimation (MLE), and forecasting.

4.1.2 Heterogeneous Auto-regressive model

Although ARFIMA models are capable of characterizing long-memory features observed in the time series of RV, their relatively poor performance led to the development and successful implementation of a class of models called Heterogeneous Autoregressive (HAR) models by Corsi (2009). In fact, relatively parsimonious structure and its success in modeling persistence in volatility and prediction (see, Patton and Sheppard , 2015; Izzeldin et al. , 2019) makes HAR model the default choice for a benchmark (see, also Christensen et al. , 2023). The HAR model is essentially a restricted $AR(22)$ model as such

$$RV_t = \beta_0 + \beta_d RV_{d,t-1} + \beta_w RV_{w,t-1} + \beta_m RV_{m,t-1} + u_t, \tag{4}$$

where $RV_{d,t-1}$ is the daily RV in the past day, and $RV_{w,t-1} = \frac{1}{5}\sum_{i=1}^{5} RV_{d,t-i}$ and $RV_{m,t-1} = \frac{1}{22}\sum_{i=1}^{22} RV_{d,t-i}$ are the weekly and monthly lagged RV, respectively. The inclusion of daily, weekly, and monthly lags of RV aims to capture the long-memory dynamic typically observed in the RV time series. Since HAR model is linear in parameters, it is estimated via Least Squares.

[4] Although one can determine AR and MA orders by using information criteria such as Bayesian Information Criterion (BIC), in this paper, we use AR and MA orders of one as this simplifies the estimation over each period. Our exploration with different AR and MA orders in the initial estimation sample covering 1996 and 2005 and in the full sample generally suggests the sufficiency of an ARFIMA(1,d,1) specification.

4.2 Nonlinear HAR Models

Since ML algorithms in general are known to model nonlinear relationship and dynamics considerably well (see the discussion in Hastie et al. , 2009; Goodfellow et al. , 2016, and references therein), it is important to consider nonlinear econometric models in order to better understand and evaluate the performances of both time series and machine learning models in a comprehensive manner. To this end, in this paper, we consider relatively simpler extensions of HAR model to capture potential nonlinear temporal dynamics along with long-range dependence. More specifically, we introduce nonlinearity into the HAR model in Equation (4) by using models where nonlinear regime or state-dependence is determined by an observable variables as well as models where state-dependence is driven by a latent variable and follows a Markov process. Under the first class of models, we consider two widely used regime-switching models one where nonlinearity in RV follows and on-and-off dynamics via an indicator function that defines the specific prevailing regime at a date t based on the deviation of a threshold variable from an estimated threshold value and another where such dynamics is modeled by the help of a smooth transition function. Under the second class of models, we consider a Markov switching dynamic regression model to capture potential nonlinearity in the HAR coefficients.

4.2.1 Threshold HAR model

The simple extension we consider is threshold type nonlinear dynamic effects. Threshold regressions, also known as threshold models or regime-switching models, are a type of econometric modeling technique that allows for a more flexible representation of relationships between variables compared to traditional linear regression models. These models are particularly useful when relationships are nonlinear and can change at certain threshold values or switching points. The key idea relative to the HAR model in Equation (4) is that the specific HAR equation parameters that characterize the temporal dynamics of RV may differ depending on the value of an additional variable, often referred to as the "threshold variable." This threshold variable acts as a signal or switch that triggers a change in the regression relationship and hence, the volatility dynamics. An excellent survey of threshold regressions is given by Hansen (2011).

In a single threshold model there is one critical threshold value of the threshold variable (z_{t-d}). When the threshold variable crosses this value (i.e., θ), it triggers a change in the regression coefficients, leading to different regression relationships in different regimes. This simple nonlinear dynamic can be characterized by writing down;

$$\begin{aligned} RV_t = &[\beta_0 + \beta_d RV_{d,t-1} + \beta_w RV_{w,t-1} + \beta_m RV_{m,t-1}] I(z_{t-d} \leq \theta) \\ &+ [\beta_0^* + \beta_d^* RV_{d,t-1} + \beta_w^* RV_{w,t-1} + \beta_m^* RV_{m,t-1}] I(z_{t-d} > \theta) + u_t, \end{aligned} \tag{5}$$

where the HAR equation parameters are changes whether an appropriate threshold variable z_{t-d}, $d = 1, \cdots, d^{max}$ is below or above the threshold value θ. d is a discrete delay parameter value that dictates the past values of z from t in driving the threshold effects. For the purpose of this paper, we do not conduct a comprehensive search for potential threshold variable but consider the relative change in RV, $z_{t-d} = \frac{\Delta RV_{t-d}}{RV_{t-d}}$ for $d = 1, \cdots, 5$ as the candidate threshold variable. In a multiple THAR model, there can be several critical threshold values. Each of these thresholds can trigger a change in the RV dynamics, resulting in multiple regimes with distinct HAR regression coefficients. Our exploratory analysis suggested either a single threshold model or a double-threshold model be the most relevant model in describing the threshold effects.[5]

We follow the procedure suggested by Gonzalo and Pitarakis (2002) for estimation and selection of the number of thresholds for each estimation sample in our data. Under this approach thresholds are estimated sequentially. We assume that up to two thresholds would be sufficient to characterize any threshold type nonlinear dynamics in RV. For a given delay d, the first threshold θ_1 is estimated assuming a model with two regions as in Equation 5 by minimizing the sum of squared errors (SSE). Conditional on the first threshold, the second threshold is estimated as the value that yields the minimum SSR over all observations excluding the first threshold. Gonzalo and Pitarakis (2002) show that thresholds estimated sequentially are asymptotically consistent. We implement this approach for $d = 1, \cdots, 5$ and choose the d that minimizes SSE (see, Hansen , 2011).

4.2.2 Smooth Transition HAR Model

STHAR model we consider is largely similar to the THAR model in Equation (5) with the key difference that a Logistic transition function drives the regime-switching dynamics. The model can be written as

$$\begin{aligned} RV_t = &[\beta_0 + \beta_d RV_{d,t-1} + \beta_w RV_{w,t-1} + \beta_m RV_{m,t-1}](1 - F(\gamma,\, \theta,\, z_{t-d})) \\ &+ [\beta_0^* + \beta_d^* RV_{d,t-1} + \beta_w^* RV_{w,t-1} + \beta_m^* RV_{m,t-1}]F(\gamma,\, \theta,\, z_{t-d}) + u_t, \end{aligned} \tag{6}$$

with the transition function, $F(.) = \frac{1}{1+\exp\{-\gamma[z_{t-d}-\theta]\}}$. The parameter θ similar to Equation (5) can be interpreted as the threshold between the two regimes corresponding to $F(\gamma,\, \theta,\, z_{t-d}) = 0$ and $F(\gamma,\, \theta,\, z_{t-d}) = 1$, in the sense that the logistic function changes monotonically from 0 to 1 as z_{t-d} increases, while $F(\gamma,\, \theta,\, z_{t-d} = \theta) = 0.5$. The parameter γ is also called the slope parameter for the transition function, determines the smoothness of the transition dynamics

[5]Note that there are different approaches one may consider to introduce threshold effects into the HAR model. In this paper, we use a simple and possibly a 'naive' approach with relatively well-known framework in the established nonlinear econometrics literature. See, for example Motegi etal. (2020) which introduces a Moving average threshold HAR model which allows time-variation in the threshold parameter θ.

from one regime to the other with large values indicating faster and potentially abrupt shift dynamics as in THAR model.

Following Leybourne etal. (1998), we use nonlinear least squares (NLS) by minimizing the concentrated sum of squares function with respect to γ and θ and similar to THAR model, use $z_{t-d} = \frac{\Delta RV_{t-d}}{RV_{t-d}}$ for $d = 1, \cdots, 5$ as the candidate transition variable and choose d with the lowest SSE from each NLS as the delay parameter. For detailed discussions of smooth transition regression and autoregressions and estimation of the model see, Granger and Terasvirta (1993), Terasvirta (1994), Leybourne etal. (1998), and Franses and van Dijk (2000).

4.2.3 Markov Switching HAR Model

Markov-switching models are another nonlinear time series models to model transition over a finite set of unobserved states. Markov switching calls of models allow the time series process to evolve differently in each state where the transitions occur according to a Markov process. The time of transition from one state to another and the duration between changes in state is random. For example, these models can be used to understand the process that governs the time at which RV transitions between episodes of high and low volatility and the duration of each episode. Detailed discussions and applications of models can be found in Quandt (1972), Goldfeld and Quandt (1973), Hamilton (1989) and Hamilton (1994). Here we provide a high-level overview in the context of extending the HAR model in Equation (4). See, Wang et al. (2020) which provides an overview of different Markov Switching HAR models proposed in the recent literature and an extension of MSHAR model with time-varying transition probabilities. Here we use a relatively simple version which essentially makes HAR parameters and the constant term to switch between two states(i.e., $s = 0, 1$ according to a Markov process:

$$RV_t = \beta_{0_{s_t}} + \beta_{d_{s_t}} RV_{d,t-1} + \beta_{w_{s_t}} RV_{w,t-1} + \beta_{m_{s_t}} RV_{m,t-1} + u_{s_t}, \tag{7}$$

where $\beta_{0_{s_t}}$ is the state-dependent intercept, $\beta_{d_{s_t}}$, $\beta_{w_{s_t}}$, and $\beta_{m_{s_t}}$ are the state-dependent coefficients, and u_{s_t} is an independent and identically distributed (i.i.d) error term with mean 0 and state-invariant variance σ^2.

If we were to know the specific state, then the MSHAR model in Equation (7) can be estimated by Least Squares by introducing s_t as a dummy variable. This is almost similar to the THAR model with the difference that the specific state is known and hence, is not driven by the deviation of a threshold variable z_{t-d} from a threshold value θ. Under MSHAR model, we assume that we do not know the specific state at any time t and hence, s_t is not observed. The state transition probabilities, $p_{s_t,s_{t+1}}$ along with the model parameters are estimated via likelihood methods. We follow Stata's Dynamic Markov Switching Regression model routine in estimating the MSHAR model. The estimation and prediction of transition probabilities are

based on the Expectations Maximization (EM) algorithm as developed by Hamilton (1989) and Kim (1994), and discussed in Hamilton (1994).

5 Machine Learning Models

ML models we entertain in this paper consist offeed-forward neural network (NN), recurrent neural networks (RNNs), as well as a tree-based model, Extreme Gradient Boosting (XGB). Our training, hyper-parameter tuning, and cross-validation approach allows us to consider NNs between a shallow network (i.e., a Multilayer Perceptron, MLP) with only one input layer to a deep neural network (DNN) with multiple hidden layers in between the input and the output layers. The specific class of RNN models we consider are neural networks that are directly suitable for modeling sequences of data in which each value is assumed to be dependent on previous values. In this class of models, we entertain four architectures allowing for a simple basic RNN (BRNN) unit to more complex architectures including Long-Short-Term Memory (LSTM), LSTM with an Attention mechanism (LSTM-A), and Gated Recurrent Unit (GRU). Excellent discussions of NN and RNN models can be found in Goodfellow et al. (2016) while Friedman (2001) provides an in-depth discussion of Gradient Boosting. Here we provide an outline of the specific architecture of each of the models used in the paper.

5.1 XGBoost

The tree-based model we use is the Extreme Gradient Boosting (XGB) which builds an ensemble of weak decision trees sequentially, where each tree corrects the errors of its predecessor. As suggested by Hastie et al. (2009) XGB is an efficient and scalable ML algorithm that can capture the potential non-linear relationships between the dependent variable and a set of predictors. It belongs to the family of gradient boosting algorithms Chen and Guestrin (2016). XGB builds an ensemble of weak learners, typically decision trees, in a sequential manner, where each new learner is trained to correct the errors made by the existing ensemble. The final prediction is obtained by aggregating the predictions of all weak learners.

XGB uses decision trees as base learners. Each decision tree is grown sequentially, optimizing a differentiable loss function such as Mean Squared Error (MSE). Trees are added to the ensemble until a predefined stopping criterion is met, such as reaching the maximum number of trees or achieving minimal improvement in the loss function, an aspect called Boosting (see, Friedman , 2001).[6]

As discussed in Chen and Guestrin (2016) XGB is trained in an additive manner, with each new tree being trained to minimize the residual errors of the ensemble. The training process

[6]Boosting means that new models are added to minimize the errors made by existing models, until no further improvements are achieved.

involves the following key steps: (i) initialize the ensemble with a simple model, typically a constant prediction or a small tree; (ii) train trees iteratively as such for each iteration, compute the negative gradient of the loss function with respect to the current predictions and then fit a decision tree to the negative gradients, optimizing a differentiable loss function; (iii) apply regularization techniques to prevent over-fitting, such as limiting the depth of trees, adding penalties to the tree weights, or subsampling the training data and features; (iv) stop adding new trees when a predefined stopping criterion is met, such as reaching the maximum number of trees or no further improvement in the loss function on a validation set; and (v) make predictions by summing the predictions of all trees in the ensemble:

$$\widehat{RV}_{t+1} = \sum_{k=1}^{K} f_k(\boldsymbol{x}_t) \tag{8}$$

where K is the number of trees in the ensemble (a hyper-parameter that is determined via time series five fold cross validation), $f_k(\boldsymbol{x}_t)$ is the prediction of the kth tree in the ensemble for the input sample vector $\boldsymbol{x}_t$.

Each tree contributes a weighted prediction to the ensemble, where the weights are determined during training and cross-validation process based on the optimization of the MSE loss function. In our grid search, we tune a number of hyper-parameters including K, learning rate, maximum depth of each decision tree as well as a number of other parameters by using a five-fold time series cross validation for each training and validation period. Our aim is to conduct a grid search that exhaustively searches through the specified parameter grid to find the combination of hyperparameters that yields the best performance, evaluated using cross-validation.

5.2 Feed-Forward Neural Networks

A feed-forward neural network (FNN) is one of the simplest types of artificial neural networks, where information moves in one direction—from input to output—without loops or feedback connections. The architecture includes an input layer, a number of hidden layers, and an output layer. The FNN learns by applying transformations at each layer using weights, biases, and activation functions. The input layer simply forwards the input features, $x_t = [x_{1t}, x_{2t}, \cdots, x_{nt}]'$, to the hidden layers ($l = 1, 2,$ ..., L) by means of an activation function, $g_l(.)$. Letting w_l, b_l and $z_t^{[l]}$ be the weight matrix, bias vector and the output of the lth hidden layer, respectively the output of the first hidden layer or input layer is given by the dot product of the input matrix with the initial weight matrix w_1 and the initial bias term b_1:

$$z_t^1 = g_1(x_t w_1 + b_1). \tag{9}$$

Output of subsequent hidden layers are given by

$$z_t^l = g_l(z_t^{l-1} w_l + b_l) \quad \text{for} \quad i = 2, 3, ..., \text{L}. \tag{10}$$

Given the output of the last hidden layer L, the output layer gives the direct prediction of our target variable as we a linear activation function for predicting the $h-$step ahead RV in the output layer,

$$\widehat{RV}_{t+h} = z_t^{[L]} w_{\text{output}} + \boldsymbol{b}_{\text{output}}. \tag{11}$$

During training, model learns the weights w^l and biases b^l by minimizing mean squared error (MSE) loss function using Adaptive Moment Estimation (Adam) optimizer (see, Kingma et al. , 2014). For the activation function, after some exploration of various alternatives, we use Leaky Rectified Linear Unit (L-ReLU) as it is able to infer when the activation is zero via gradient-based methods (see, Christensen et al. , 2023). The L-ReLU is defined as

$$\sigma(x) = \alpha \cdot x, \text{if}\, x < 0$$
$$x, \quad \text{otherwise}$$

where $\alpha \geq 0$. We follow the literature and set $\alpha = 0.01$.

As the discussion above suggests, a NN essentially applies a number of functional transformations to build a forecast in the output layer. The Universal Approximation Theorem of Cybenko (1989) implies that a single hidden layer with a large enough number of units and an appropriate activation function should be sufficient to approximate any continuous function. In contrast, state of the art NNs structures considers adding extra hidden layers than to arbitrarily increasing the number of neurons in a layer. Our model training, hyper-parameter tuning and cross validation approach as discussed in Appendix A allows the data decide/choose the main network structure in terms number of layers and number of units within each layer, ranging from only one layer with relatively large number of neurons to deep networks with multiple layers and neurons in each layer.

5.3 Recurrent neural networks

Given that the temporal dynamics and the persistence in realized volatility time series data, a potentially more relevant class of neural networks is the RNNs which are designed to model sequential data in which each value is assumed to be dependent on previous values. More specifically, RNNs similar to NNs are feed-forward networks augmented by a feedback loop (see, Goodfellow et al. , 2016). As explained in Chung et al. (2014) and Goodfellow et al. (2016), RNNs introduce the notion of time to the otherwise standard feed-forward NNs and hence,

should be able to model temporal dynamics. In this paper, we consider four RNNs including a basic RNN unit (referred as BRNN in the paper), Long Short-Term Memory (LSTM), LSTM with an Attention mechanism (LSTM-A), and a simplified version of LSTM, called Gated Recurrent Unit (GRU). See, Chung et al. (2014), and Lipton et al. (2015) for a comprehensive review and comparison of different RNN architectures. Here we provide a short overview of the four models we entertain in the paper.

5.3.1 Basic RNN

BRNN is a simple RNN unit and allows model's output from the previous period to be an an additional input to the model at time step t. More specifically, under a BRNN unit, output of the model at period t in a given layer is a function of the previous output and the current input and can be described by

$$z_t = \tanh(x_t w_1 + z_{t-1} w_2 + b) \tag{12}$$

where z_t is the output at time t, x_t is the model's input time series consisting of T samples, with w_1, w_2 and b are the model's parameters (weights), and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is the hyperbolic tangent function. As the Equation (12) illustrates, the output from the previous period $t-1$ z_{t-1} is an additional input to the model at time t, along with the current input x_t. The hyperbolic tangent activation function essentially allows the basic RNN unit to model nonlinear relationship between the past value of the output, the current input and the output of the model.

5.3.2 Long-short-term memory

Although basic RNN architecture is suitable to model temporal dynamics, it may not capture the long-range dependence especially when the time series data is long enough as it uses data from recent history to forecast (see, Hochreiter and Schmidhuber , 1997). In other words, BRNN model may not carry relevant information from earlier periods in the sequence to later ones, such as specific patterns from the same week in past year. LSTM networks mitigate this so called, "short-term memory" problem by introducing gates that enable the preservation of relevant information from the past and hence, holding the long-term memory and combining it with the most recent data (Hochreiter and Schmidhuber , 1997). In other words, in an LSTM, layers receive their input from both the input layer of the current time step and the layer from the previous time step, thereby allowing the network to have a memory of past events. The LSTM and LSTM-A models have been successfully implemented in applications spanning handwriting recognition, speech recognition, handwriting generation, machine translation, image captioning, and parsing (see, Graves et al. , 2013; Graves and Jaitly , 2014; Graves , 2013; Sutskever et al. , 2014; Xu et al. , 2015; Vinyals etal. , 2014).

At a high level, an LSTM network (and hence also, an LSTM with attention mechanism) consists of one or more LSTM layers stacked together. One of the key layers is the input layer which receives sequential data, organized into sequences with multiple time steps where each time step may contain multiple features. Each LSTM layer processes the sequential input data and captures temporal dependencies. To achieve this, typically each LSTM layer consists of multiple LSTM cells or units arranged along the time axis. At each time step, an LSTM cell processes the input and updates internal states, including memory cell state (c_t) and hidden state h_t. Computations within an LSTM cell involves input, forget, and output gates (or vectors), and memory cell updates. The output layer receives the final hidden states or outputs from LSTM layers and may consists of one or more dense layers for further processing or directly outputs the model's prediction.

By construction, an LSTM cell has the ability to 'memorize' or 'forget' information through the use of a special memory cell state, carefully regulated by three gates: an input gate, a forget gate, and an output gate. The gates regulate the flow of information into and out of the memory cell state. These gates regulate the flow of information as such input gate regulates the flow of new information into the memory cell c_t, forget gate regulates the retention or forgetting of information from the previous memory cell state (c_{t-1}, and output gate regulates the exposure of information from the current memory cell state (c_t) as the output of the LSTM cell.

The transformation in each cell of the LSTM layer can be defined as follows. The input gate takes the current input time series data x_t, the previous hidden state vector h_{t-1} as inputs and computes how much of the new information (i_t) should be stored in the current memory cell sate (c_t) via the equation using

$$i_t = \sigma(x_t w_1^i + h_{t-1} w_2^i + b^i), \tag{13}$$

where i_t is the input cell/gate output, w_1^i, and w_2^i are the weight parameters corresponding to the input time series data at time step t and the hidden state from the previous time step, ($h_[t-1]$), respectively and b^i is the bias vector. The activation function, $\sigma(.) = \frac{1}{1+e^x}$ is the sigmoid activation function to produce values between 0 and 1. These values represent how much of the new information should be stored in the memory state.

The forget gate takes the current input data and the previous hidden state and produces values between 0 and 1 indicating how much of the previous memory cell state should be retained in the current time step t;

$$f_t = \sigma(x_t w_1^f + h_{t-1} w_2^f + b^f), \tag{14}$$

where f_t is the forget gate output vector and w_1^f, w_2^f, and b^f are corresponding weight and bias parameters. The output gate is defined similarly and provides how much of the memory cell

should be exposed as the output of the LSTM cell.

$$o_t = \sigma(x_t w_1^o + h_{t-1} w_2^o + b^o), \tag{15}$$

where o_t is the output gate's output with the corresponding weight and the bias parameters as in the previous equations. Together these gates allow LSTM cells to selectively process and retain relevant information over time, enabling effective long-term dependency modeling in time series data.

In addition to above equations characterizing the flow of information in input, forget, and output gates, an important component of LSTM cell is called the candidate cell state or candidate activation for the cell state which is determined by the current time data input x_t and the previous period's hidden state via the equation:

$$\tilde{c}_t = \tanh(x_t w_1^c + h_{t-1} w_2^c + b^c), \tag{16}$$

where $tanh(.)$ is the hyperbolic tangent function which ensures that the new candidate state is bounded between -1 and 1 and carries information from the current input data and the information from the previous hidden state. In the LSTM cell, the input gate determines how much of this candidate cell state should be incorporated into the memory cell state, allowing the LSTM cell to learn long-term dependencies while mitigating the vanishing gradient problem (see, Goodfellow et al., 2016). Given the memory cell state and the candidate cell states at time step t, the new memory cell state at the current time step t is obtained via

$$c_t = f_t \circ .c_{t-1} + i_t \circ \tilde{c}_t \tag{17}$$

where $\circ$ is the element-wise product operator (Hadamard product). This equation essentially indicates how the outputs of input and forget gates define the current cell state as the input gate i_t determines which parts of the candidate $\tilde{c}$ should be used to modify/update the memory cell state, and forget gate f_t determines which parts of the previous memory c_{t-1} should be discarded in forming the current cell state. Given this recently updated cell state c_t, it is 'squashed' through the nonlinear hyperbolic tangent function first and then the output gate o_t determines which parts of it should be presented in as the output of the hidden state h_t at time-step t via

$$h_t = o_t \circ \tanh(c_t). \tag{18}$$

5.3.3 LSTM with Attention

The third RNN model we consider is an LSTM model with an attention mechanism (LSTM-A). The foundational work for understanding and implementing LSTM models with attention

mechanisms is Bahdanau etal. (2015). The LSTM-A model architecture we implement and the associated LSTM cell processing equations are the same as in Equations 13 through 18 above. The key difference between this model and the LSTM model above is that we include an attention mechanism with the objective to improve LSTM's ability to focus on relevant parts of our time series data in making predictions. The attention mechanism takes the outputs of the LSTM layers as input and it computes the attention scores between the LSTM output and itself. The resulting attention weights are used to compute the weighted sum of the LSTM output, which serves as the attention output. We also include a dense layer after the attention mechanism with the Leaky ReLU activation function to provide additional flexibility and expressiveness to the model, allowing it to learn more complex relationships between the features and our target; potentially improving its performance. Additionally, similar to DNN and LSTM models, we add a dropout layer for the purpose of regularization. The final output layer uses linear activation function in computing the model's forecast as in DNN and LSTM models.

5.3.4 Gated Recurrent Unit

The fourth RNN model we use is the Gated Recurrent Unit (GRU) which potentially improves the LSTM model by dropping the *cell* state in favor of a more simplified architecture that requires fewer learn-able parameters (Dey and Salemt , 2017). The key difference between LSTM models and GRUs is that GRUs employ only two gates instead of three to control the flow of information, namely the update gate and the reset gate. Since GRUs are parsimonious relative to LSTM and LSTM-A models, they are considered to be faster and more efficient, especially when training data are limited (see, Dey and Salemt , 2017). Broadly in line with the equations for an LSTM cell, the following set of four equations define a GRU unit:

$$u_t = \sigma(x_t w_1^u + h_{t-1} w_2^u + b^u) \tag{19}$$

$$r_t = \sigma(x_t w_1^r + h_{t-1} w_2^r + b^r)$$

$$v_t = \tanh(x_t w_1^v + (h_{t-1} \times r_t) w_2^v + b^v)$$

$$h_t = u_t \circ v_t + (1 - u_t) h_{t-1}, \tag{20}$$

where w_1^u, w_2^u, and b^u are the weight and bias parameters that control the *update gate* u and w_1^r, w_2^r, and b^r are the parameters that control the reset gate r. The update and reset gate equations in 21 determine the flow of information to be fed into the candidate activation at time-step t, v_t, and subsequently to the hidden-state output h_t. The candidate activation v_t is a function of the input data at time-step t x_t and the output of h_{t-1} and is controlled

by parameters, w_1^v, w_2^v, and b^v. The output h_t combines the candidate activation v_t and the previous output state h_{t-1} as controlled by the update gate u_t.

5.4 Training, validation, and testing approach

Our model training, validation, and testing approach follows splitting the data into training and validation and testing samples starting with the initial training and validation sample of the first ten years of sample for the period 01/02/1996 and 12/31/2005 and using the subsequent year of sample for out-of-sample testing. We repeat this by expanding the training and validation sample by one year and moving the out-of-sample testing period accordingly. We use a time series cross validation procedure which divides the training/validation sample into five contiguous train and test folds (i.e. each training set grows in size by adding previous folds with each split) while using a grid search for hyper-parameter tuning. In addition to several key hyper-parameters, we also use this approach in letting the data deciding the model architecture in terms of number of layers and neurons within each layer. For econometric models, we use the training and validation sample to estimate the parameters and use the estimated models for prediction in the out-of-test sample.[7]

For the purpose of evaluating the performance of models and comparing their performances in the out-of-sample periods since 2006, we use Mean Squared Prediction Error (MSPE), Quasi-likelihood of Patton and Sheppard (2009) as our loss measures. We utilize the model confidence set of Hansen et al. (2011) in conjunction with DM test of equal predictive ability of Diebold and Marioano (1995). We use the realized utility approach of Bollerslev et al. (2018) to evaluate the economic benefit of using predictions under a model. Finally, we use likelihood ratio tests for unconditional and conditional coverage tests to assess models in their ability to predict market risk in terms of Value at Risk (VaR) under a filtered historical simulation. We also compare the accuracy of VaR predictions under a loss function proposed by Gonzalez-Rivera et al. (2004). The detailed discussions of our training, validation, hyper-parameter tuning, and testing are provided in Appendix A. As discussed in Appendix A, our training, validation, and hyper-parameter tuning approach allows the data for each training and validation sample period to define the specific architecture of each ML models in addition to key hyper-parameters for ML models.

6 Empirical Results

For brevity, we report and discuss our main empirical findings in this section based on the baseline scenario, where the predictor set includes only the past history of realized volatility (i.e.,

[7] Given the constraints in terms of re-training and validating ML models, we follow this annual re-training and validation and testing approach 2006 on-wards instead of more frequent training and validation and testing.

Table 1: ML models key architecture parameters over each training/validation sample

Sample end	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
2005	3[50], 3[50]	4, 1	4, 1	1, 1	5, 1	1, 5
2006	3[50], 3[50]	1, 1	1, 5	5, 1	5, 1	4, 1
2007	3[50], 3[50]	5, 1	1, 5	1, 1	1, 1	1, 1
2008	3[50], 3[50]	4, 1	1, 5	1, 1	4, 1	4, 1
2009	3[200], 3[100]	5, 1	5, 1	1, 1	5, 1	1, 5
2010	3[200], 3[50]	1, 1	1, 1	5, 1	1, 1	5, 1
2011	3[50], 3[50]	5, 1	1, 5	1, 1	5, 1	4, 1
2012	3[50], 3[100]	5, 1	1, 5	1, 1	4, 1	5, 1
2013	3[50], 3[100]	1, 1	1, 1	1, 1	5, 1	1, 5
2014	3[50], 3[200]	5, 1	1, 1	1, 1	4, 1	5, 1
2015	3[200], 3[100]	4, 1	5, 1	1, 1	5, 1	1, 1
2016	3[100], 3[100]	4, 1	5, 1	1, 1	5, 1	4, 1
2017	5[50], 3[50]	1, 1	1, 1	1, 1	1, 1	4, 1
2018	3[50], 3[50]	5, 1	5, 1	1, 1	1, 1	1, 1
2019	3[50], 3[200]	5, 1	1, 1	3, 1	1, 1	1, 1
2020	3[50], 3[200]	5, 1	1, 1	1, 1	1, 1	1, 1
2021	3[50], 3[50]	4, 1	1, 1	1, 1	5, 1	1, 5
2022	3[50], 3[100]	5, 1	1, 1	1, 1	1, 1	1, 1

Notes: The sample end column reports the training and validation sample ending at the given year end since 1996. The values for XGB column are the selected maximum tree depth and the number of decision trees or boosting rounds (in squared brackets) while values for DNN and RNNs are the number of hidden layers and the corresponding number of neurons selected by our training/validation and hyper-parameter tuning approach. For XGB, the potential maximum tree depth and the number of boosting rounds considered are $3, 5, 7, 10$ and $5, 10, 25, 50, 100, 200, 300$, respectively. For DNN and RNNs, the possible choices of number of layers and neurons considered are $1 : [32]$, $2 : [4, 2]$, $3 : [8, 4, 2]$, $4 : [16, 8, 4, 2]$, $5 : [32, 16, 8, 4, 2]$. First values under each model column are the selected key elements under the baseline scenario with HAR set of predictors while the second are under the additional set of predictors.

the set of predictors suggested by the HAR model). Findings based on the extended predictor set—which includes additional macroeconomic and financial market variables alongside the HAR predictors—are presented in Appendix B.

Additionally, results for recurrent neural networks (RNNs), including BRNN, GRU, LSTM, and LSTM-A, trained with a longer predictor set (i.e., 22 lags of RV) are provided in Appendix C as robustness checks. Contrary to our expectations, these results indicate that extending the predictor set to include up to 22 lags of past RV generally leads to poorer performance compared to using only the HAR set of predictors. Given this, we focus our main analysis on the baseline scenario, where only the HAR predictor set is used.

Before delving into a detailed discussion of our findings, Table 1 presents the specific architecture of the ML models selected through our data-driven training, validation, and hyper-parameter tuning approach, which is described in detail in Appendix A. As outlined in the

appendix, our approach treats key elements of ML model architecture as hyper-parameters and conducts a grid search over a predefined set of values. For DNN and RNN models, these hyper-parameters include the number of hidden layers and the number of units within each layer. For XGB, they include the maximum depth of each tree and the total number of decision trees.

For each training and validation sample leading up to the out-of-sample test period, Table 1 reports these key architecture elements for both predictor sets: (i) the HAR predictor set (first entries under each model column in the table) and (ii) the HAR predictor set plus additional macroeconomic and financial market predictors (second entries under each model column).

An inspection of the results reveals that, unlike the common practice in recent RV literature of fixing model architecture, our data-driven approach identifies significant variations in architecture both across training and validation samples for a given model and across models. Interestingly, model architectures appear relatively more uniform when additional predictors are included. Nevertheless, the results highlight the importance of a data-driven approach, as evolving information arrival can influence the optimal architecture of ML models.[8]

In presenting our empirical results in this Section, we first discuss performance of models in predicting one-day ahead RV, then the economic utility of using such forecasts in portfolio allocation, and then finally the performance of models in predicting VaR.

6.1 Predicting realized volatility

Panels A and B of Table 2 present average prediction errors as measured by MSPE and QLIKE, respectively for each model across all test years since 2006 and the entire test sample period, between 2006 and 2023. Results in the last rows of Panel A and B for the entire test period indicate that the top five models in terms of attaining the lowest prediction errors under both MSPE and QLIKE are STHAR, HAR, THAR, MSHAR, and DNN. The clear winner in terms of achieving the lowest MSPE and QLIKE is the STHAR model. MCS procedure finds STHAR as the only model in the MCS in the entire test sample period while none are excluded under QLIKE.[9]

Careful inspection of reported MSPE and QLIKE across test periods in the Table and the displayed plots in Figure 1 reveal that HAR and especially its nonlinear variants, THAR and STHAR tend to have lower or similar average prediction errors across test periods under both loss measures compared to ML models. ARFIMA model tends to perform the worst across all test periods with the exception of test period 2008. Inspection of reported results in Table 2

[8]The findings on model architecture warrant further investigation, which we leave for future research. Additionally, for brevity, we do not report parameter estimates for ARFIMA, HAR, and its nonlinear extensions (THAR, MHAR, and STHAR), though these results are available upon request.

[9]As discussed in Hansen et al. (2011), when the MCS procedure leads outcomes where all models are in the MCS, this may imply that the procedure is not informative and hence, a complementary predictive accuracy testing approach could be used which we also undertake by using DM tests.

Table 2: One-day ahead MSPE and QLIKE over test periods

YEAR	HAR	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTAM-A
Panel A. One-day ahead MSPEs											
2006	0.012	0.013	0.013	0.012	0.018	0.013	0.012	0.013	0.012	0.012	0.013
2007	0.047	0.047	0.054	0.049	0.063	0.050	0.049	0.048	0.048	0.048	0.075
2008	0.313	0.354	0.415	0.322	0.446	0.588	0.374	0.556	0.541	0.492	0.512
2009	0.041	0.045	0.061	0.040	0.058	0.044	0.042	0.047	0.044	0.043	0.049
2010	0.073	0.065	0.076	0.072	0.088	0.074	0.075	0.073	0.074	0.074	0.072
2011	0.082	0.069	0.088	0.082	0.098	0.089	0.079	0.073	0.081	0.080	0.076
2012	0.019	0.018	0.020	0.019	0.027	0.020	0.019	0.019	0.019	0.019	0.019
2013	0.022	0.022	0.022	0.020	0.031	0.022	0.022	0.023	0.021	0.021	0.022
2014	0.023	0.020	0.025	0.023	0.030	0.026	0.023	0.025	0.023	0.024	0.024
2015	0.088	0.076	0.090	0.079	0.110	0.067	0.087	0.086	0.087	0.085	0.081
2016	0.028	0.028	0.032	0.026	0.040	0.028	0.028	0.029	0.029	0.028	0.029
2017	0.008	0.008	0.009	0.008	0.011	0.010	0.009	0.008	0.009	0.011	0.010
2018	0.059	0.053	0.065	0.061	0.075	0.067	0.058	0.061	0.058	0.059	0.058
2019	0.026	0.026	0.027	0.025	0.038	0.028	0.026	0.026	0.027	0.027	0.028
2020	0.177	0.221	0.215	0.173	0.265	0.242	0.191	0.183	0.182	0.224	0.203
2021	0.044	0.042	0.045	0.042	0.055	0.043	0.042	0.043	0.043	0.044	0.043
2022	0.073	0.069	0.084	0.071	0.100	0.073	0.076	0.075	0.075	0.073	0.080
2023	0.022	0.027	0.024	0.021	0.034	0.022	0.022	0.022	0.022	0.022	0.022
06-23	0.064	0.067	0.076	0.045	0.089	0.085	0.070	0.079	0.079	0.078	0.080
Panel B. One-day ahead QLIKE											
2006	0.006	0.006	0.007	0.006	0.009	0.006	0.006	0.007	0.006	0.006	0.007
2007	0.025	0.025	0.032	0.026	0.036	0.026	0.026	0.026	0.026	0.026	0.042
2008	0.330	0.552	0.774	0.332	1.008	2.267	0.583	1.563	1.447	1.237	1.296
2009	0.022	0.023	0.037	0.021	0.032	0.024	0.021	0.023	0.022	0.022	0.023
2010	0.057	0.050	0.071	0.057	0.098	0.059	0.063	0.066	0.059	0.058	0.065
2011	0.044	0.038	0.059	0.044	0.065	0.045	0.045	0.043	0.045	0.045	0.045
2012	0.010	0.009	0.011	0.010	0.014	0.010	0.010	0.010	0.010	0.010	0.010
2013	0.011	0.011	0.012	0.010	0.016	0.011	0.011	0.012	0.011	0.011	0.011
2014	0.012	0.010	0.013	0.012	0.016	0.013	0.012	0.013	0.012	0.012	0.012
2015	0.130	0.093	0.173	0.112	0.280	0.070	0.134	0.139	0.130	0.148	0.127
2016	0.015	0.015	0.018	0.014	0.022	0.015	0.015	0.015	0.015	0.015	0.015
2017	0.004	0.004	0.004	0.004	0.005	0.005	0.004	0.004	0.004	0.005	0.005
2018	0.034	0.030	0.042	0.035	0.047	0.037	0.032	0.032	0.032	0.032	0.032
2019	0.014	0.014	0.015	0.013	0.020	0.014	0.014	0.014	0.014	0.014	0.014
2020	0.127	0.228	0.218	0.118	0.279	0.180	0.119	0.132	0.124	0.203	0.172
2021	0.022	0.022	0.025	0.022	0.030	0.022	0.022	0.022	0.022	0.022	0.022
2022	0.042	0.038	0.053	0.040	0.061	0.041	0.041	0.041	0.041	0.041	0.046
2023	0.012	0.014	0.013	0.011	0.018	0.011	0.012	0.012	0.011	0.011	0.011
06-23	0.051	0.066	0.088	0.038	0.117	0.162	0.066	0.123	0.115	0.109	0.111

Notes: The table reports the 1-day ahead out-of-sample Mean Squared Prediction Error (MSPE) and Quasi-Likelihood (QLIKE)loss for each model for each of the test samples over the period 2006 and 2023. The last row reports MSPE and QLIKE for the entire test period of 2006-2023. Values in blue indicate models that are included in the MCS at the 5% significance level. Years in which all models are in MCS are indicated in red. Values in black indicate models that are not in the MCS.

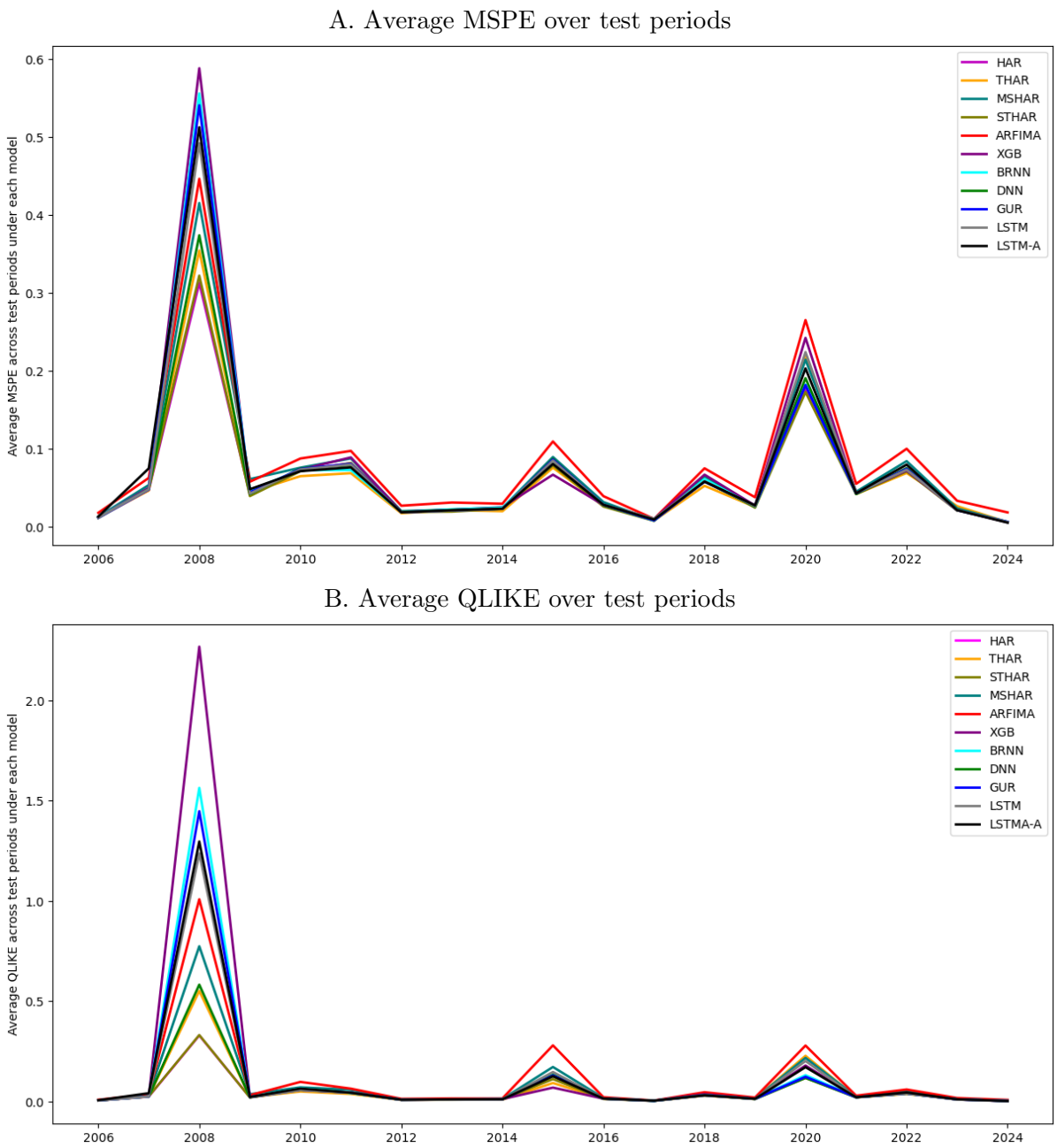
and the plots in Figure 1 demonstrate both ML and econometric models tend to have higher MPSE and QLIKE during the market volatility episode of 2008 and periods of elevated RV such as 2010, 2015, 2020, and 2022 with larger increases in MSPE and QLIKE relative to the initial test year of 2006. As models trained with additional data, MSPE and QLIKE decline especially from the levels reached in 2008 despite episodes of elevated volatility in the recent test periods. The decline in MSPE and especially in QLIKE is more pronounced among the ML models as sharper declines in MSPE and QLIKE are observed especially if one compares the market turbulent periods of 2008 to 2020. For example, the average QLIKE declines from 1.447 to 0.124 between 2008 and 2020 for GRU, roughly a 12-fold decline as can also be glanced from the inspection of Figure 1.

It is worth noting that GRU, LSTM and LSTM-A models tend to display larger average prediction errors in 2008 compared to DNN, possibly due to their retention of more past RV information, similar to ARFIMA, and consequently failing to capture rapid RV shifts. This contrasts with nonlinear time series models like THAR and STHAR, which adjust relatively quickly to regime shifts. Note also that STHAR model performs better than THAR and MSTAR models across all test periods and in the full sample, as possibly due to the fact that it better captures the gradual and smooth shifts in RV than these other nonlinear models. HAR model performs generally the second best model after the STHAR model with its' parsimonious structure allowing it better reflect the movements in RV compared to both ML models and its nonlinear versions of THAR and MSHAR. Note also that MSHAR model does not display the same degree of performance as the threshold and smooth transition extensions of HAR model possibly due to the fact that THAR and STHAR utilizes information relatively quickly as relative shifts (acceleration or deceleration of RV beyond certain threshold values) in the RV provide useful information about the regime changes and hence, the improvement in the predictive ability. We also note that GRU quickly learns and improves its performances in 2015 and 2020 as it enables 'memorization' of relevant information patterns with significantly fewer parameters compared to LSTM and LSTM-A.

The MCS procedure indicates that STHAR model is the only model to remain in confidence set across all test periods. Indeed, for several of the test years, STHAR is the only model in MCS including 2007, 2016-2019, and 2021-2023 under MSPE and 2006, 2007, 2014, 2017-2019, and 2022 under QLIKE. The procedure leads to outcomes where all models are part of the MCS in test periods with usually elevated volatility such as 2008, 2010, 2015, and 2020. Although this outcome might possibly due to the insufficiency of the information in periods of heightened volatility in distinguishing models, the discussion below on the results from DM predictive accuracy tests provide additional clarity in these heightened volatility periods. Despite some variation across remaining test years, the same set of models, including HAR, THAR, STHAR, MSHAR, XGB, DNN, GRU, and LSTM, usually remain in the final MCS. Models with long-

range dependence including ARFIMA, LSTM, and LSTM-A tend to exhibit larger prediction errors across most test years, despite the fact that they remain within MCS for most test years with the exception of 2008, 2015, and 2020.

Figure 1: Average MSPE and QLIKE across test periods for each model



This figure displays average MSPE and QLIKE over each test year since 1996. See, the main text for description and calculation of MSPE and QLIKE.

To gain further insights into the relative performance of models, Tables 3 and 4 report pairwise DM test results by using MSPE and QLIKE as loss functions, respectively for selected test years and for the full-test period between 2006 and 2023. Each cell in these tables corresponds to the DM test between the model in that row and the model in the column. The null hypothesis, $H_0 : \ell_i = \ell_j$, for models i and j is tested against the one-sided alternative $H_1 : \ell_i > \ell_j$, where ℓ represents the loss metric (MSPE or QLIKE). A star in each entry indicates rejection of the null hypothesis, and a positive DM statistic suggests that the column model's loss measure is, on average, statistically significantly higher than that of the row model. Conversely, a negative entry indicates the opposite.

Reported DM test results in panel G of Table 3 reveal that STHAR is the clear outperformer, boasting not only smaller MSPEs than all other models but also statistically superior performance in terms of achieving the lowest MSPE in the entire test sample. Additionally, HAR and THAR models outperform MSHAR and ARFIMA and all ML models. DNN although tends to have higher average MSPE compared to HAR and THAR, statistically it performs equally well vis a vis with THAR and outperforms MSHAR. XGB performs poorly relative to all neural networks and all econometric models except for ARFIMA under MSPE in the full-test sample. ML models that are suitable for long-range memory, outperforms XGB, and especially ARFIMA model and display a similar performance overall.

Upon examining the DM test results under QLIKE in Panel H of Table 4, notable differences in statistical significance emerge compared to the results under MSPE in panel G of Table 3. While the signs of the test statistics remain consistent with the results under MSPE, the statistical significant differences between pair of models largely disappear. For instance, although HAR and STHAR models still exhibit lower QLIKE against all models, the DM test results do not reveal any meaningful statistical differences in QLIKE across models especially between HAR and THAR and any of the ML models. This result in a way confirms our finding from the MCS procedure in the full-test period under QLIKE which found no statistically meaningful differences among models and hence, all models were found to be in the MCS. STHAR model continues to pioneer in terms of its' performance and displays statistically significant lower average QLIKE against all other econometric models as well as against DNN, LSTM, and LSTM-A. Interestingly enough, performances of XGB and GRU are statistically indistinguishable from that of STHAR under QLIKE under 5 percent significance level. Additionally DM tests do not suggest dominance of any ML model over others under the QLIKE, contrary to the reported results under MSPE in panel G of Table 3 in Panel A of the Table. Overall, DM tests under MSPE and QLIKE show that HAR model and specifically STHAR model outperforms competitors with generally statistically significant prediction errors under both measures of loss functions with some variation in the relative performances of ML models across these two measures.

Table 3: Diebold-Mariano statistics under MSPE over selected test periods

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
A. Test Period 2008										
HAR	-0.94	-1.99*	-0.53	-1.98*	-3.02*	-2.47*	-2.52*	-2.63*	-2.94*	-2.49*
THAR		-1.80	-0.09	-2.16*	-3.28*	-2.37*	-2.61*	-2.81*	-3.22*	-2.67*
MSHAR			1.18	-0.83	-3.37*	-2.50*	-3.62*	-3.67*	-3.88*	-3.04*
STHAR				-1.75	-3.45*	-1.57	-2.12*	-2.47*	-3.13*	-2.38*
FI					-3.32*	0.05	-1.16	-1.97*	-3.15*	-1.96
XGB						3.02*	3.03*	2.91*	2.22*	3.18*
DNN							-2.35*	-2.73*	-3.31*	-2.15*
BRNN								-3.32*	-3.75*	-1.55
GRU									-3.93*	-0.13
LSTM										3.46*
B. Test Period 2010										
HAR	1.16	-0.45	2.14*	-0.92	-0.22	-0.48	-0.01	-0.54	-0.35	0.15
THAR		-1.28	2.11*	-1.24	-1.17	-1.74	-1.04	-1.73	-1.59	-0.87
MSHAR			1.94	-1.09	0.41	0.23	0.96	0.29	0.37	1.14
STHAR				-1.76	-2.21*	-2.13*	-1.89	-2.23*	-2.24*	-1.84
ARFIMA					0.98	0.84	1.19	0.83	0.86	1.24
XGB						-0.34	0.09	-0.22	-0.05	0.27
DNN							0.55	0.33	0.51	0.73
BRNN								-0.24	-0.12	0.76
GRU									1.05	0.41
LSTM										0.30
C. Test Period 2015										
HAR	1.04	-0.18	1.42	-0.89	1.23	0.10	0.36	0.58	0.32	0.77
THAR		-0.74	1.53	-1.00	1.06	-1.05	-0.86	-1.01	-0.70	-0.53
MSHAR			1.17	-1.28	0.90	0.23	0.48	0.32	0.79	0.87
STHAR				-1.25	-1.48	-1.39	-1.31	-1.40	-1.19	-1.24
ARFIMA					1.07	0.94	1.07	0.96	1.23	1.19
XGB						-1.18	-1.04	-1.18	-0.89	-0.84
DNN							0.53	0.93	0.40	0.90
BRNN								-0.15	0.27	0.87
GRU									0.22	0.80
LSTM										0.72
D. Test Period 2020										
HAR	-1.53	-1.62	2.07*	-2.75*	-1.85	-1.27	-0.91	-0.77	-1.92	-1.62
THAR		0.32	2.21*	-2.19*	-0.44	0.93	1.47	1.35	-0.11	0.94
MSHAR			2.35*	-2.74*	-0.67	0.76	1.39	1.25	-0.48	0.74
STHAR				-2.84*	-2.48*	-2.30*	-2.22*	-2.17*	-2.74*	-2.46*

Continued on next page

Table 3: (continued)

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
ARFIMA					0.57	2.03*	2.74*	2.54*	1.72	2.90*
XGB						1.54	1.79	1.82	0.47	1.12
DNN							0.74	1.26	-1.11	-0.57
BRNN								0.21	-1.77	-1.45
GRU									-1.60	-1.22
LSTM										1.91
E. Test Period 2022										
HAR	1.40	-2.44*	3.56*	-4.54*	0.33	-0.93	-0.86	-0.81	-0.14	-1.68
THAR		-2.47*	3.26*	-4.01*	-0.93	-1.90	-1.97*	-1.96	-1.57	-2.34*
MSHAR			3.92*	-3.31*	2.24*	1.40	1.66	1.64	1.94	0.70
STHAR				-4.88*	-3.57*	-3.79*	-3.79*	-3.82*	-3.71*	-3.70*
ARFIMA					4.79*	3.87*	3.98*	3.93*	4.26*	2.94*
XGB						-1.40	-1.06	-1.00	-0.49	-1.93
DNN							0.72	0.72	1.51	-1.42
BRNN								0.18	2.68*	-1.51
GRU									2.45*	-1.44
LSTM										-1.79
F. Test Period 2023										
HAR	-1.46	-0.94	1.75	-5.42*	0.69	-0.52	0.25	1.59	0.80	1.10
THAR		0.79	2.30*	-1.70	1.58	1.39	1.46	1.56	1.54	1.61
MSHAR			1.75	-6.04*	0.98	0.73	1.09	1.22	0.96	1.07
STHAR				-5.05*	-1.57	-1.92	-1.68	-1.58	-1.69	-1.57
ARFIMA					5.21*	5.42*	5.55*	5.61*	5.22*	5.21*
XGB						-1.16	-0.47	-0.06	-0.44	0.17
DNN							0.59	1.42	1.01	1.49
BRNN								1.07	0.36	0.71
GRU									-0.49	0.29
LSTM										1.09
G. Full Test Period 2006-2023										
HAR	-0.90	-3.67*	6.96*	-5.60*	-2.89*	-1.82	-2.18*	-2.69*	-2.81*	-3.08*
THAR		-3.11*	6.13*	-5.72*	-2.85*	-0.15	-0.65	-2.38*	-2.63*	-2.94*
MSHAR			7.38*	-5.10*	-1.58	3.60*	3.06*	-0.54	-0.50	-1.08
STHAR				-8.10*	-5.84*	-7.10*	-7.23*	-6.26*	-6.48*	-6.78*
ARFIMA					0.84	5.68*	5.51*	2.73*	3.39*	2.81*
XGB						2.94*	2.85*	2.00*	1.86	1.50
MLP							-2.49*	-2.85*	-2.88*	-3.30*
DNN								-2.75*	-2.69*	-3.21*
GRU									0.28	-0.89

Continued on next page

Table 3: (continued)

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
LSTM										-1.68

Notes: The table reports DM test for the equality of MSPE between the model on the row and the model on the column over selected test periods and the full test period between 2006 and 2023. The null hypothesis being tested is $H_0 : E(\ell_{RV_i}) = E(\ell_{RV_j})$ against $H_0 : E(\ell_{RV_i}) > E(\ell_{RV_j})$, where model i is the label of the selected row, whereas model j is the label of the selected column. A * indicates rejection of the null against the one-sided alternative under the 5% significance. A positive value indicates that the average QLIKE over the test period of the model on the row is greater than the model on the column.

To understand if models' predictive ability differs across episodes of heightened and benign market conditions and in test periods where MCS procedure did not provide a clear subset of models to be the superior performers, we present DM test results under MSPE for 2008, 2010, 2011, 2015, 2020, and the most recent test years of 2022, and 2023. Clearly test periods of 2008 and 2020 marked by significant market upheaval, 2010 and 2011, and 2022 with relatively elevated volatility periods and 2023 with relatively benign market conditions.[10] Inspection of the results in Panel A of Table 3 show that HAR and its nonlinear variants, THAR and STHAR, outperform ML models by producing statistically lower average MSPE during this period of extreme market volatility. Specifically, STHAR model beats all models by achieving statistically lower average MSPE in 2008. Among the ML models, XGB performs the worst while DNN outperforms other ML models and LSTM beats GRU and LSTM-A.

Learning from the experience of market volatility of 2007-2008, ML models display improvement in 2020 as statistical significance of DM statistic disappears when HAR and THAR models are compared with any of the ML models as can be seen by comparing the DM test results in panels A and D of Table 3. STHAR again stands out as the top performer, achieving statistically lower MSPE in both 2008 and 2020 against all models. This suggests that despite the significant improvements made by the ML models, a nonlinear HAR model still can beat ML models as it potentially captures the time series dynamic well with a relatively parsimonious structure when the set of predictors include only the HAR variables. During 2020, ML models generally perform the same against each other despite the fact that GRU achieves much smaller MSPE compared to other ML models. Pairwise DM predictive accuracy test results for 2008 and 2020 under MSPE also show that DM test can provide useful insights in distinguishing model's predictive accuracy even in periods of elevated volatility, a result in contrast to MCS procedure which tend to include all models in the confidence set.

Pair-wise DM tests in Table 3 for test years 2010 and 2015, two test years MCS procedure included all models in the MCS, shows that STHAR model attains statistically lower MSPE compared to majority of the models in 2010 while none of the models statistically stand out

[10]Results for all other test periods under both MSPE and QLIKE are available in an online Appendix.

in 2015. In 2010, relatively high volatility period spanning the European Sovereign Crisis, STHAR achieves statistically significant smaller MSPE than all ML models at 5% or 10% significance levels. Interestingly enough STHAR has statistically lower MSPE than HAR and THAR models in 2010 but statistically indistinguishable from MSHAR and ARFIMA models at 5% significance level.

Inspection of the reported results for 2022 in Table 3, a period of elevated volatility, shows that similar to 2020, STHAR model outperforms all models by attaining statistically lower MSPE against all models. ARFIMA model remains the worst performer against all models in this period. Similar to 2020, THAR model outperforms HAR and MSHAR models in 2022. In contrast to 2020, THAR now achieves statistically lower MSPE against BRNN and LSTM-A but continue to have average MSPE statistically indistinguishable from those of XGB, DNN, GRU and LSTM. Similar to 2020, the performance of DNN generally remains to be indistinguishable from that of RNN models in 2022.

In the test period 2023, where overall market is relatively calmer compared to 2008 and 2020, the differences in the performances of models diminish and almost disappear as in 2015 with a few exceptions, despite the fact that STHAR continues to produce lower MSPE albeit non-significant DM test results. This result is potentially driven by the continued improvements in the ability of ML models with the expansion of the data sample covering the period between 1996 and 2022 in the training stage as these models presumably foster in terms of predictive performance in data-rich environments. Results under MSPE generally suggest that in calmer market conditions, the statistical significance of predictive accuracy test tends to disappear as most models tend to perform equally well with the exception of ARFIMA. The differences between models sharpen as market volatility increases with larger and sudden increases in RV and THAR and especially STHAR outperform most models. In test periods where MCS procedure includes all models in the MCS, we find that pair-wise DM test provides additional insights into the performance of models either by narrowing down the set as in 2011 or by confirming the outcome of MCS as in 2015. This latter observation might be driven by how elevated the volatility in the period under investigation.

Reported results in Table 4 under QLIKE loss although qualitatively similar in terms of the sign of the DM statistics across model pairs, the statistical differences between models diminish and in many cases disappear especially in 2008 and 2020. Notably STHAR model outperform statistically DNN model in 2008 while ARFIMA model outperforms statistically LSTM and LSTM-A models. Despite some differences in QLIKE, pair-wise DM tests confirm results of MCS procedure for 2008. This confirmation continues to hold for 2010 as none of the models stand out statistically. These results are different from the findings we have under MSPE, underscoring the importance of loss function used under DM tests.

In 2011, again a period of relatively elevated volatility, pair-wise DM tests provide relatively strong results. For example, statistically THAR model outperforms MSHAR, but outperformed by HAR which is beaten by STHAR. Inspection of results in Panel C in the Table reveals that 2011 is a year where STHAR achieves statistically significant lower average QLIKE compared to most models at 5% significance level. In 2015, however, results are similar to the ones we have under MSPE as no model stands out in terms of achieving statistically significant lower QLIKE, confirming the findings from MCS procedure.

In contrast, in 2022, the DM test under QLIKE indicates statistically significant differences among several models. For example, HAR, THAR and STHAR models as well as most ML models outperform MSHAR and ARFIMA models and STHAR is the only model beating all ML models in 2022, this is despite the relatively notable improvements in prediction errors noted in ML models in 2022 compared to say 2008 as reported in Table 2. Results in the last panel of Table 4 show that the noted differences among models tend to diminish under QLIKE in 2023, a test year with relatively low levels of RV.

Table 4: Diebold-Mariano statistics under QLIKE over selected test periods

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
A. Test Period 2008										
HAR	-1.05	-1.43	-0.06	-1.23	-1.21	-1.64	-1.37	-1.41	-1.38	-1.41
THAR		-2.09*	1.48	-1.33	-1.24	-0.38	-1.46	-1.53	-1.53	-1.55
MSHAR			1.78	-0.96	-1.16	1.21	-1.34	-1.39	-1.33	-1.38
STHAR				-1.38	-1.26	-2.51*	-1.48	-1.53	-1.53	-1.55
ARFIMA					-1.20	1.06	-1.58	-1.78	-2.02*	-2.02*
XGB						1.16	1.00	1.01	1.09	1.06
DNN							-1.31	-1.35	-1.30	-1.34
BRNN								1.06	1.33	1.25
GRU									1.52	1.41
LSTM										-1.92
B. Test Period 2010										
HAR	0.85	-1.21	1.41	-1.06	-0.97	-1.27	-1.10	-0.88	-0.48	-1.07
THAR		-1.14	1.56	-1.04	-0.90	-1.25	-1.09	-1.18	-1.12	-1.11
MSHAR			1.35	-0.98	1.21	1.00	1.23	1.09	1.11	1.11
STHAR				-1.18	-1.39	-1.45	-1.35	-1.48	-1.48	-1.36
ARFIMA					1.06	0.98	1.01	1.01	1.02	1.00
XGB						-0.89	-1.04	0.08	0.37	-0.96
DNN							-0.75	1.22	1.25	-0.69
BRNN								0.97	1.01	0.64
GRU									1.32	-0.95

Continued on next page

Table 4: (continued)

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
LSTM										-1.00
C. Test Period 2011										
HAR	2.02*	-1.66	3.00*	-1.75	-0.07	-0.08	0.62	-0.60	-0.19	-0.04
THAR		-2.03*	2.80*	-2.03*	-1.17	-2.18*	-1.52	-2.26*	-2.18*	-2.08*
MSHAR			2.49*	-1.23	1.29	1.68	2.02*	1.63	1.67	1.77
STHAR				-2.39*	-2.92*	-3.00*	-2.78*	-3.06*	-3.02*	-3.05*
ARFIMA					1.38	1.76	2.01*	1.72	1.75	1.81
XGB						0.05	0.29	0.00	0.05	0.06
DNN							0.77	-0.46	-0.08	0.02
BRNN								-0.88	-0.80	-1.34
GRU									1.34	0.22
LSTM										0.05
D. Test Period 2015										
HAR	0.99	-0.97	1.06	-1.01	1.02	-0.96	-0.91	-0.57	-0.91	1.04
THAR		-0.98	1.13	-1.00	1.05	-0.99	-0.98	-0.99	-0.97	-0.95
MSHAR			1.03	-1.02	1.00	0.97	0.99	0.98	1.02	1.01
STHAR				-1.03	-1.26	-1.06	-1.05	-1.06	-1.03	-1.04
ARFIMA					1.01	1.01	1.02	1.01	1.02	1.02
XGB						-1.02	-1.00	-1.01	-0.99	-0.99
DNN							-0.87	1.04	-0.89	1.16
BRNN								0.93	-0.91	1.04
GRU									-0.92	1.14
LSTM										0.99
E. Test Period 2020										
HAR	-1.58	-2.08*	1.55	-1.90	-1.07	0.82	-0.74	0.39	-1.75	-1.62
THAR		0.42	1.76	-1.31	0.75	1.57	1.61	1.60	0.96	1.49
MSHAR			2.08*	-1.40	0.76	1.96	2.12*	2.07*	0.88	2.32*
STHAR				-1.89	-1.32	-1.19	-1.51	-1.33	-1.96	-1.78
ARFIMA					1.76	1.92	1.96	1.96	1.45	1.94
XGB						1.31	1.08	1.24	-0.41	0.19
DNN							-1.23	-0.93	-1.68	-1.59
BRNN								1.50	-1.74	-1.70
GRU									-1.73	-1.69
LSTM										1.57
F. Test Period 2022										
HAR	1.87	-3.45*	3.44*	-3.63*	0.38	0.19	0.82	1.01	1.32	-1.66
THAR		-3.21*	3.30*	-3.37*	-1.17	-1.34	-1.61	-1.52	-1.39	-2.33*

Continued on next page

Table 4: (continued)

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
MSHAR			3.89*	-2.23*	3.18*	2.80*	3.01*	3.00*	3.18*	1.93
STHAR				-4.01*	-3.43*	-2.56*	-3.13*	-3.19*	-3.33*	-2.77*
ARFIMA					3.76*	3.01*	3.03*	3.02*	3.02*	1.91
XGB						-0.92	1.08	1.04	1.00	1.03
DNN							-3.15*	-3.00*	-3.13*	-3.15*
BRNN								-2.04*	-3.29*	-3.01*
GRU									3.18*	2.04*
LSTM										3.17*
G. Test Period 2023										
HAR	-1.36	-1.58	1.69	-5.01*	0.78	-0.67	-0.29	0.75	1.08	0.94
THAR		0.44	2.18*	-1.52	1.51	1.26	1.33	1.40	1.46	1.49
MSHAR			1.93	-5.63*	1.48	1.28	1.72	1.71	1.55	1.55
STHAR				-4.55*	-1.50	-1.95	-1.64	-1.60	-1.61	-1.56
ARFIMA					4.71*	5.02*	5.16*	5.14*	4.82*	4.76*
XGB						-1.51	-0.71	-0.58	-0.44	-0.23
DNN							0.52	1.17	1.40	1.60
BRNN								0.77	0.81	0.79
GRU									0.55	0.71
LSTM										0.47
H. Full Test Period 2006-2023										
HAR	-1.19	-2.07*	2.17*	-1.96*	-1.21	-1.62	-1.40	-1.40	-1.52	-1.51
THAR		-2.84*	2.89*	-2.21*	-1.21	0.15	-1.42	-1.43	-1.61	-1.60
MSHAR			3.15*	-1.77	-0.99	2.36*	-1.01	-0.93	-0.97	-0.99
STHAR				-2.50*	-1.42	-3.71*	-1.78	-1.84	-2.05*	-2.02*
ARFIMA					-0.75	2.03*	-0.29	0.10	0.79	0.49
XGB						1.17	0.97	1.02	0.99	0.98
DNN							-1.34	-1.34	-1.48	-1.47
BRNN								1.30	1.04	1.02
GRU									0.79	0.69
LSTM										-0.88

Notes: The table reports DM test for the equality of QLIKE between the model on the row and the model on the column over selected test periods and the full test period between 2006 and 2023. The null hypothesis being tested is $H_0 : E(\ell_{RV_i}) = E(\ell_{RV_j})$ against $H_0 : E(\ell_{RV_i}) > E(\ell_{RV_j})$, where model i is the label of the selected row, whereas model j is the label of the selected column. A * indicates rejection of the null against the one-sided alternative under the 5% significance. A positive value indicates that the average QLIKE over the test period of the model on the row is greater than the model on the column.

Table 5: Realized Utility over test periods

Year	HAR	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
2006	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2007	0.039	0.039	0.039	0.040	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2008	0.031	0.023	0.016	0.030	0.007	-0.039	0.022	-0.012	-0.008	-0.001	-0.003
2009	0.040	0.040	0.039	0.040	0.039	0.040	0.040	0.040	0.040	0.040	0.040
2010	0.039	0.039	0.038	0.039	0.037	0.039	0.038	0.038	0.039	0.039	0.038
2011	0.039	0.039	0.039	0.040	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2012	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2013	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2014	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2015	0.036	0.037	0.035	0.039	0.031	0.038	0.036	0.036	0.036	0.035	0.036
2016	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2017	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2018	0.039	0.039	0.039	0.040	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2019	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2020	0.037	0.034	0.034	0.038	0.032	0.036	0.037	0.037	0.037	0.035	0.036
2021	0.040	0.040	0.039	0.040	0.039	0.040	0.040	0.040	0.040	0.040	0.040
2022	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2023	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
06-023	0.039	0.038	0.038	0.039	0.037	0.035	0.038	0.036	0.037	0.037	0.037

Notes: The table reports the realized utility benefit of using forecasts of volatility based on different models under the assumptions of a constant conditional Sharpe ratio equal to 0.40 and a coefficient of risk aversion $\gamma = 2$. The maximum utility benefit in this setting using the future realized volatilities, is equal to 4

6.2 Realized utility

The previous section illustrated that HAR, particularly its nonlinear extension using a transition function like an indicator or a 'smooth' logistic function, generally outperforms ML models in terms of attaining lower prediction errors, especially during periods of heightened market volatility. In this subsection, we assess to what extent econometric and ML models' RV predictions are useful in informing an investor forming a portfolio as captured by the realized utility approach described by Bollerslev et al. (2018). Recall that under this approach a perfect model delivers a realized utility of 4% and estimated realized utility values below 4% would indicate less accurate prediction by a given model.

Table 5 reports realized utility measures for each model across each test period and in the entire test period between 2006 and 2023. Reported results in the last row for the entire test period show that none of the models delivers exact 4% realized utility but econometric models, namely HAR and STHAR achieves a realized utility of 3.9% while THAR, MSHAR, and DNN delivers 3.8% estimated realized utility. Among the eighteen test periods investigated, STHAR model delivers the perfect realized utility in thirteen test periods, while HAR, THAR and all ML models in ten test periods. MSHAR and ARFIMA models deliver perfect realized utility estimates in only eight of the test periods. Generally models fail to deliver 4% realized utility in periods where the realized volatility is elevated. Notably in 2008, all models result in smaller

than 4% realized utility with all ML models resulting in negative realized utility. These results are broadly in line with the predictive accuracy test results and the reported average prediction errors in the previous section.

6.3 Predicting VaR

Volatility forecasting serves various purposes, including informing the prediction of return distribution in portfolios and quantifying tail risk, crucial for risk management, regulatory reporting, and determining initial margins. In this section, we assess linear and nonlinear econometric models and ML models' performance in predicting 1-day ahead VaR by using the FHS method discussed in Section A.2. To this end, we use two approaches one that assesses predicted VaRs under each model in terms of their backtesting performance and two that evaluates predictive accuracy of model pairs via DM test under the loss criteria in Equation 26.

We first present the exceedance/failure rate, the ratio of log returns falling below predicted VaR relative to the number of VaR days in a test sample, and test results for correct unconditional and conditional coverage rates. The Kupiec test (Kupiec , 1995) assesses whether the failure rate for a given model matches the VaR confidence level, testing unconditional coverage. The second test, by Christoffersen (1998), evaluates whether exceedances in a VaR period are independent and have correct coverage. We calculate VaR at both 95 and 99 confidence levels but report results at 99 percent level for brevity.

Table 6 presents results from the first approach, reporting exceedance rates under each model across test years from 2006 to 2023 and the entire sample period, 1996-2023. An asterisk indicates the rejection of the null of correct coverage rate and a dagger that of correct conditional coverage at five percent significance level. Reported failure rates in the last row for the entire sample period show that all models fail to achieve expected coverage rates, with STHAR displaying the lowest failure rate at 1.4 percent.

Looking at the test periods individually shows that STHAR consistently achieves both conditional and unconditional coverage, with exceedance rates statistically indistinguishable from the one percent VaR confidence level. Only exception to this result is 2007, where STHAR model displays a higher failure rate of 2.8 percent. Failure rates by all models increase during periods of heightened volatility, including the GFC of 2007-2008, and test periods of 2011, 2015, and the recent volatility episodes of 2020 and 2022. HAR, THAR, and MSHAR models, perform overall similarly in terms of attained exceedance rates and periods of incorrect coverage. ARFIMA performs the worst among both all models, with ten test periods of statistically significant incorrect unconditional and conditional coverage results.

ML models generally perform similarly in terms of failure rates and coverage capabilities over time, with roughly around five to six periods of statistically significant test periods of incorrect conditional and/or unconditional coverage. As the training sample covers data from

Table 6: VaR Exceedance Rates & Kupiec and Christoffersen Tests over test periods and the entire test sample

Year	HAR	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
2006	0.016	0.024	0.016	0.008	0.020	0.012	0.016	0.016	0.020	0.012	0.028*†
2007	0.040*†	0.044*†	0.048*†	0.028*	0.048*†	0.048*†	0.048*†	0.044*†	0.040*†	0.040*†	0.028*
2008	0.028*	0.040*†	0.051*	0.020	0.036*†	0.032*†	0.040*†	0.056*†	0.056*†	0.052*†	0.056*†
2009	0.016	0.012	0.020	0.012	0.028*	0.016	0.016	0.020	0.016	0.016	0.016
2010	0.028*	0.032*†	0.036*†	0.016	0.040*†	0.032*†	0.028*	0.036*†	0.020	0.028*	0.036*†
2011	0.036*†	0.036*†	0.044*†	0.020	0.048*†	0.036*†	0.036*†	0.036*†	0.036*†	0.036*†	0.036*†
2012	0.016	0.016	0.016	0.016	0.020	0.016	0.016	0.016	0.016	0.016	0.016
2013	0.028*	0.024	0.028*	0.020	0.032*†	0.028*	0.028*	0.032*	0.028*	0.028*	0.028*
2014	0.032*†	0.028*	0.036*†	0.020	0.036*†	0.032*†	0.032*†	0.036*†	0.032*†	0.028*	0.032*†
2015	0.036*†	0.032*†	0.032*†	0.008	0.036*†	0.028*†	0.036*†	0.036*†	0.036*†	0.028*†	0.028*†
2016	0.016	0.012	0.012	0.008	0.016	0.008	0.016	0.016	0.012	0.012	0.012
2017	0.016	0.012	0.016	0.012	0.016	0.016	0.016	0.016	0.016	0.016	0.016
2018	0.028*†	0.024	0.024	0.004	0.028*†	0.024	0.024	0.024	0.020	0.024	0.024
2019	0.016	0.020	0.024	0.012	0.024	0.008	0.012	0.012	0.012	0.012	0.008
2020	0.036*†	0.036*†	0.044*†	0.020	0.040*†	0.032*†	0.028*†	0.028*†	0.028*†	0.032*†	0.032*†
2021	0.016	0.016	0.028	0.000	0.028*	0.020	0.020	0.020	0.016	0.016	0.016
2022	0.036*†	0.028*†	0.036*†	0.016	0.032*†	0.028*	0.024	0.032*†	0.028*	0.028*	0.036*†
2023	0.017	0.017	0.021	0.008	0.021	0.017	0.017	0.017	0.017	0.017	0.017
06-23	0.025*†	0.025*†	0.029*†	0.014*†	0.030*†	0.024*†	0.025*†	0.027*†	0.025*†	0.024*†	0.026*†

Notes: The table reports VaR Exceedance Rates over each test year and in the entire test period between 2006 and 2023 across models/ The exceedance rate is defined to be the number of days in a test period where log returns falls below the predicted VaR divided by the number of VaR days in a period. A * indicates the rejection of the null hypothesis of correct unconditional coverage (i.e.., rejection of the null by the Kupiec test) at 5 percent significance level that the exceedance rate equals to 1 percent (i.e., the VaR confidence level of 99 percent) against the one-sided alternative that it is more than 1 percent. A †indicate the rejection of the correct conditional coverage rate at 5% significance level. The last row gives results for the entire test period, 2006-2023.

volatility episodes, both econometric and ML models' performance in achieving correct coverage rates generally improves with enhanced predictive ability of RV. For instance, DNN, and GRU exhibit lower exceedance rates with fewer rejections of correct unconditional and conditional coverage, especially in 2020 and 2022 compared to earlier such episodes.

Table 7 reports the results of the DM test for pairs of models under the loss function described in Equation 26, aimed at evaluating the predictive accuracy of VaR forecasts. We reports results for a selected test periods in addition to the the entire test sample period for brevity as results are qualitatively similar acros majority of the test periods examined. Analysis of the results in the last panel, Panel E of the Table, indicates that while the HAR model achieves lower loss compared to all ML models, its predictive accuracy does not show statistical distinction from any of the ML models. THAR demonstrates a similar performance, except for outperforming XGB at five percent level. MSHAR outperforms only ARFIMA model in terms of achieving lower VaR loss while outperformed by all other models. STHAR emerges as the clear front-runner again, exhibiting significantly lower VaR loss compared to all other models. XGB exhibits statistically higher VaR loss relative to all neural networks. Despite its' poorer performance against all ML models, XGB outperforms MSHAR, and ARFIMA models with statistically significant lower VaR loss. DNN achieves lower VaR loss albeit insignificant DM test against all other ML models. The performances of GRU, LSTM, and LSTM-A show no statistically distinguishable differences among them.

Panels A-D of Table 7 further illustrate model performance during heightened volatility periods in 2008, 2020, 2022 and in recent calmer period of 2023. These results indicate that econometric models, specifically HAR and its' two nonlinear variants although generally achieve lower average loss compared to ML models in the high volatility episode of 2008, DM test does not suggest statistically significant lower loss between these models and ML models. STHAR remains the best performing model in terms of achieving lower average loss differential relative to all models and larger DM test (in absolute value) outcomes in both volatility episodes of 2008 and 2020. Consistent with the improvement in their ability to predict RV, ML models improve especially in 2020 compared to 2008 episodes as evidenced by the change in the sign of the DM statistic against HAR model and statistically significant DM test results against MSHAR and ARFIMA models in 2020 relative to 2008. Despite the overall improvement in predictive ability of all models, STHAR model outperforms all models especially in 2020 relative to 2008. In the relatively elevated volatility period of 2022, THAR model performs equally well compared to HAR and STHAR and outperforms ML models with generally statistically significant lower average VaR loss. In the most recent year with relatively calmer volatility, pairwise-DM tests suggest no statistically significant loss differential across any of the models. Similar results hold for other periods of low vs. high volatility episodes. Full results are available in an online Appendix.

Table 7: DM tests of predictive accuracy of VaR over selected test periods

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
A. Test Period 2008										
HAR	-0.09	-1.71	1.16	-1.57	-0.88	-1.21	-1.47	-1.62	-1.42	-1.53
THAR		-1.18	0.77	-1.07	-0.79	-0.77	-1.26	-1.47	-1.25	-1.37
MSHAR			1.88	-0.15	0.03	1.22	-0.57	-0.99	-0.56	-0.83
STHAR				-1.9	-1.21	-1.62	-1.92	-1.99*	-1.86	-1.93
ARFIMA					0.07	1.05	-0.43	-0.83	-0.42	-0.67
XGB						0.25	-0.44	-0.83	-0.4	-0.6
DNN							-1.18	-1.45	-1.11	-1.36
BRNN								-1.68	-0.11	-1.05
GRU									2.19*	1.62
LSTM										-2.43*
B. Test Year 2020										
HAR	-0.50	-1.96	2.17*	-1.98*	-1.5	1.04	-0.05	0.26	-0.65	-0.79
THAR		-1.60	1.9	-2.2*	-1.72	0.72	0.6	0.62	0.05	-0.05
MSHAR			2.62*	-1.86	0.23	1.99*	2.02*	1.98*	2.06*	2.13*
STHAR				-2.61*	-2.42*	-2.18*	-2.18*	-2.21*	-2.18*	-2.27*
ARFIMA					2.21*	2.03*	2.06*	2.03*	2.35*	2.31*
XGB						1.62	1.64	1.61	2.08*	1.99*
DNN							-1.06	-1.26	-0.94	-1.11
BRNN								0.72	-0.84	-1.04
GRU									-0.84	-1.02
LSTM										-0.40
C. Test Year 2022										
HAR	1.94	-1.74	1.83	-1.19	0.42	0.67	0.89	0.94	1.16	-2.10*
THAR		-1.97*	1.44	-1.69	-1.95	-2.3*	-2.31*	-2.42*	-2.01*	-2.27*
MSHAR			2.04*	0.38	1.68	1.61	1.65	1.66	1.74	1.22
STHAR				-2.05*	-1.84	-1.95	-1.95	-1.95	-1.92	-2.0*
ARFIMA					1.31	1.38	1.33	1.37	1.39	0.46
XGB						0.66	0.69	0.91	0.95	-1.72
DNN							0.09	0.68	0.45	-1.55
BRNN								0.5	0.57	-1.71
GRU									0.15	-1.69
LSTM										-1.82
D. Test Year 2023										
HAR	1.18	-0.11	1.91	-1.2	-1.14	-0.39	1.94	-0.38	0.04	-0.58
THAR		-0.70	1.39	-1.22	-1.40	-1.12	-0.83	-1.18	-1.27	-1.27
MSHAR			1.40	-1.45	-0.43	-0.16	0.46	-0.07	0.11	-0.32
STHAR				-1.95	-1.85	-1.57	-1.81	-1.63	-1.70	-1.57
ARFIMA					1.01	0.98	1.29	1.09	1.08	0.96
XGB						0.40	1.54	1.05	1.09	0.16

Table 7 continued from previous page

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
DNN							0.88	0.31	0.69	-0.41
BRNN								-1.02	-0.91	-0.99
GRU									0.56	-0.86
LSTM										-0.89
E. Full Test Period 2006-2023										
HAR	0.86	-4.97*	6.71*	-5.65*	-1.75	-0.95	-1.64	-1.38	-1.16	-1.44
THAR		-4.70*	5.72*	-5.86*	-2.34*	-1.21	-1.89	-1.67	-1.75	-1.93
MSHAR			7.10*	-3.99*	2.36*	4.59*	3.13*	3.02*	3.8*	3.19*
STHAR				-7.63*	-6.13*	-6.9*	-7.09*	-6.59*	-6.51*	-6.64*
ARFIMA					4.54*	5.32*	4.54*	4.32*	5.13*	4.71*
XGB						1.30	0.65	0.62	0.99	0.56
DNN							-1.50	-1.23	-0.85	-1.26
BRNN								0.06	0.57	-0.20
GRU									0.61	-0.26
LSTM										-1.13

Notes: The table reports Diebold-Mariano test of equal predictive accuracy of predicted VaR under VaR loss function for the test year 2008. The null hypothesis being tested is $H_0 : E(\ell_{VaR_i}) = E(\ell_{VaR_j})$ against $H_0 : E(\ell_{VaR_i}) > E(\ell_{VaR_j})$, where model i is the label of the selected row, whereas model j is the label of the selected column. A * indicates rejection of the null against the one-sided alternative under the 5% significance. A positive value indicates that the average loss under the model on the row is greater than the model on the column.

7 Conclusions

In this paper, we have examined the predictive performance of both traditional econometric models and machine learning (ML) algorithms in forecasting realized volatility (RV) for the S&P 500 index using high-frequency data. Specifically, we compared benchmark econometric models, including ARFIMA and HAR, with a set of advanced ML models such as Extreme Gradient Boosting (XGB), deep neural networks, and recurrent neural networks (RNNs), including BRNN, LSTM, GRU, and LSTM with attention mechanisms (LSTM-A). Our analysis aimed to assess whether the nonlinearity inherent in financial time series can be more effectively captured by ML techniques or through regime-switching extensions of econometric models, such as THAR, MSHAR, and STHAR.

Our results provide several key insights. First, we show that simple yet robust econometric models, when extended to incorporate regime-switching dynamics, can compete with and often surpass more complex ML models in terms of forecasting accuracy. This finding challenges the

assumption that ML models inherently outperform econometric models, particularly when predictor sets are limited to past volatility values. The success of threshold and smooth transition HAR models highlights the importance of accounting for regime changes in financial markets, which can have significant implications for predicting market volatility across varying economic conditions.

Second, while many recent studies have emphasized the benefits of large-scale predictor sets and sophisticated ML architectures, our results suggest that careful model specification and fitting schemes, such as retraining and re-estimation frequency, play a crucial role in enhancing model performance. In fact, the HAR model and its' nonlinear extension when applied with refined fitting strategies, remains highly competitive against more computationally intensive models like RNNs. This reinforces the value of econometric models for practitioners and researchers seeking parsimonious yet effective tools for volatility forecasting.

Third, we extend the literature by employing a dynamic training and validation process that allows the architecture of the neural networks to be selected by the data, ensuring that the models evolve over time as new data and information become available. This approach contrasts with the more static architectures typically used in the existing literature, adding a novel dimension to how ML models can be applied in financial forecasting.

Finally, our study offers important practical implications for the application of ML and econometric models in financial markets. While ML techniques are powerful tools, they require careful consideration of hyperparameters, fitting schemes, and predictor sets, especially when applied to persistent time series data like RV. Moreover, our findings indicate that nonlinear extensions of econometric models, such as regime-switching HAR variants, can serve as strong contenders, offering a more interpretable and parsimonious approach to modeling volatility dynamics.

Future research could extend our framework by exploring alternative high-frequency datasets, other volatility measures, and different market environments to assess the generalizability of our findings. Additionally, investigating hybrid models that combine the strengths of econometric and ML approaches may yield further improvements in volatility forecasting, especially in capturing the nonlinearities and regime dependencies present in financial markets.

References

Alexander, Carol (2009). Market Risk Analysis, Value at Risk Models. John Wiley & Sons.

Andersen, Torben G., Tim Bollerslev, Francis X. Diebold, and Heiko Ebens (2001). The Distribution of Realized Stock Return Volatility, *Journal of Financial Economics* 61: 43–76.

Andersen, Torben G., Tim Bollerslev, Francis X. Diebold, and Paul Labys (2003). Modeling and Forecasting Realized Volatility. *Econometrica* 71: 579–625.

Aruoba, S.B., Diebold, F.X. and Scotti, C. (2009). Real-Time Measurement of Business Conditions, *Journal of Business and Economic Statistics* 27:4, pp. 417-27.

Audrino, F., & Knaus, S. D. (2016). Lassoing the HAR Model: A Model Selection Perspective on Realized Volatility Dynamics, *Econometric Reviews,* 35, 1485–1521.

Audrino, F., Sigrist, F., & Ballinari, D. (2020). The impact of sentiment and attention measures on stock market volatility, *International Journal of Forecasting,* 36, 334–357.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR).*

Baillie, R. T. (1996). Long memory processes and fractional integration in econometrics, *Journal of Econometrics* 73: 5–59.

Baker, S. R., N. Bloom, S. Davis, J. Steven (2016). Measuring economic policy uncertainty. *Q. J. Econ.* 131 (4), 1593–1636

Barkan, O, J. Benchimo, I. Caspi, E. Cohen, A. Hammer, and N. Koenigstein (2023). Forecasting CPI inflation components with Hierarchical Recurrent Neural Networks, *International Journal of Forecasting* 39. 1145-1162.

Barndorff-Nielsen, Ole E., and Neil Shephard (2002). Econometric Analysis of Realized Volatility and Its Use in Estimating Stochastic Volatility Models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64: 253-280.

Barone-Adesi, G., K. Giannopoulos, and L. Vosper (1999). Var without correlations for portfolios of derivative securities. *J. Futures Markets* 19 (5), 583–602.

Bauwens, L., Hafner, C., & Laurent, S. (2012). *Volatility Models and Their Applications*, Volume 15. John Wiley & Sons, Inc.

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroscedasticity. Journal of Econometrics 31: 307-327.

Bollerslev, T., Hood, B., Huss, J., & Pedersen, L. H. (2018). Risk Everywhere: Modeling and managing volatility, *Review of Financial Studies,* 31, 2730–2773.

Branco, R.R., A. Rubesam, M. Zevallos (2024). Forecasting realized volatility: Does anything beat linear models?, *Journal of Empirical Finance*, 78, 1-22.

Bucci, A. (2020) Realized Volatility Forecasting with Neural Networks *Journal of Financial Econometrics* 18: 502–531.

Buckman, S.R, A. H. Shapiro, M. Sudhof, and D. Wilson (2020). News sentiment in the time of COVID-19. *FRBSF Econ. Lett.* 8 (1), 5–10.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 785-794).

Christensen, K., M. Siggaard, and B. Veliyev (2023). A Machine Learning Approach to Volatility Forecasting, *Journal of Financial Econometrics,* Vol. 21. 1680-1727.

Christoffersen, P. (1998). Evaluating Interval Forecasts *International Economic Review* Vol. 39, pp. 841–862.

Christoffersen, P. and D. Pelletier (2004). Backtesting value-at-risk: A duration-based approach. *J. Empir. Finance* 2, 84–108.

Chung, J., C. Gulcehre, K. Cho, and Y. Bengio (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv, https://arxiv.org/abs/1412.3555.

Corsi, F. (2009). A Simple Approximate Long-Memory Model of Realized Volatility, *Journal of Financial Econometrics* 7: 174–196.

Corsi, Fulvio, Renó, Roberto (2012). Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling. *J. Busines and Economic. Statistics* 30 (3), 368–380.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function *Math. Control Signal Systems* 2, 303–314.

Díaz J.D., E. Hansen, G. Cabrera (2024). Machine-learning stock market volatility: Predictability, drivers, and economic value, *International Review of Financial Analysis,* 94 103286.

Dey, R., & Salemt, F. M. (2017). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (pp. 1597–1600). IEEE.

Diebold, F.X. and R.S. Mariano. (1995). Comparing Predictive Accuracy, *Journal of Business and Economic Statistics,* 13: 253-63

Donaldson, G. R., and M. Kamstra (1997). An Artificial Neural network-GARCH Model for International Stock Return Volatility, *Journal of Empirical Finance* 4: 17–46.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica 50: 987-1007.

Fama, E., and K. French. (1993). Common Risk Factors in the Returns on Stocks and Bonds, *Journal of Financial Economics* 33: 3–56.

Fernandes, M., M. C. Medeiros, and M. Scharth (2014). Modeling and Predicting the CBOE Market Volatility Index, *Journal of Banking & Finance* 40: 1–10.

Franses, P. H. and D. van Dijk (2000). Non-Linear Time Series Models in Empirical Finance. Cambridge University Press.

Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine, *The Annals of Statistics* Vol. 29, pp. 1189-1232.

Giot, Pierre, Laurent, S., (2004). Modelling daily value-at-risk using realized volatility and ARCH type models. *J. Empir. Finance* 11 (3), 379–398.

Goldfeld, S. M., and R. E. Quandt (1973). A Markov model for switching regressions, *Journal of Econometrics* 1: 3–15. https://doi.org/10.1016/0304-4076(73)90002-X.

Gonzalez-Rivera, G., T.-H, Lee, and S. Mishra (2004). Forecasting volatility: A reality check based on option pricing, utility function, value-at-risk, and predictive likelihood, *International Journal of Forecasting,* 20(4), 629–645.

Granger, Clive W. J. & T. Terasvirta (1993). Modelling Non-Linear Economic Relationships, Oxford University Press, New York.

Gunnarsson, E. S., Isern, H. R., Kaloudis, A., Risstad, M., Vigdel, B., & Westgaard, S. (2024). Prediction of realized volatility and implied volatility indices using AI and machine learning: A review, *International Review of Financial Analysis,* 93, 103221.

Hamid, S. A., and Z. Iqbal (2004). Using Neural Networks for Forecasting Volatility of S&P 500 Index Futures Prices, *Journal of Business Research* 57: 1116–1125.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle, *Econometrica* 57: 357–384. https://doi.org/10.2307/1912559.

Hamilton, J. D. (1994). Time Series Analysis. Princeton, NJ: Princeton University Press.

Hansen, P. R., and A. Lunde (2005). A Forecast Comparison of Volatility Models: Does Anything Beat a GARCH(1,1)? *Journal of Applied Econometrics* 20: 873–889.

Hansen, P. R., A. Lunde, and J. M. Nason (2011). The model confidence set, *Econometrica*, Vol 79, 453-497.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Science & Business Media.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770-778).

Hill, Tim, Marcus O'Connor, and William Remus (1996). Neural Network Models for Time Series Forecasts *Management Science* 42: 1082–1092.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, **9**(8), 1735-1780.

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). Multilayer Feedforward Networks Are Universal Approximators *Neural Networks* 2: 359–366.

Hu, Y. M., and C. Tsoukalas (1999). Combining Conditional Volatility Forecasts Using Neural Networks: An Application to the EMS Exchange Rates, *Journal of International Financial Markets, Institutions & Money* 9: 407–422.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning, MIT Press, http://www.deeplearningbook.org.

Gonzalo, J., and J.-Y. Pitarakis (2002). Estimation and model selection based inference in single and multiple threshold models. Journal of Econometrics 110: 319–352. https://doi.org/10.1016/S0304-4076(02)00098-2

Graves, A., Mohamed, A., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In ICASSP 2013 , pages 6645–6649

Graves, A. (2013). Generating sequences with recurrent neural networks. Technical report, arXiv:1308.0850.

Graves, A. and Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In ICML 2014

Hansen, B. E. (2000). Sample splitting and threshold estimation. Econometrica 68: 575–603. https://doi.org/10.1111/1468-0262.00124.

Hansen, B. E. (2011). Threshold autoregression in economics. Statistics And Its Interface 4: 123–127. https://doi.org/10.4310/SII.2011.v4.n2.a4

Hillebrand, E. and M. C. Medeiros (2010). The Benefits of Bagging for Forecast Models of Realized Volatility. *Econometric Reviews* 29 (5-6), 571-593.

Izzeldin, Marwan, M. Kabir Hassan, Vasileios Pappas, and Mike Tsionas (2019). Forecasting Realised Volatility Using ARFIMA and HAR Models, *Quantitative Finance* 19: 1627–1638.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networksm https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Kim, C.-J. (1994). Dynamic linear models with Markov-switching, *Journal of Econometrics* 60: 1–22.

Kingma, Diederik P., and Jimmy Ba (2014). Adam: A Method for Stochastic Optimization, Working paper.

Koenker, R., and G. Bassett (1978). Regression Quantiles, *Econometrica* 46: 33–50.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1097-1105).

Kupiec, P. (1995). Techniques for Verifying the Accuracy of Risk Management Models *Journal of Derivatives* Vol. 3, pp. 73 – 84.

Leybourne, S., P. Newbold and D. Vougas (1998) Unit Roots and Smooth Transitions *Journal of Time Series Analysis,* 19, 83-97.

Liu, Lily Y., Andrew J. Patton, and Kevin Sheppard (2015). Does Anything Beat 5-Minute RV? A Comparison of Realized Measures across Multiple Asset Classes, *Journal of Econometrics* 187: 293–311.

Liu, F., Pantelous, A. A., & von Mettenheim, H. J. (2018). Forecasting and trading high frequency volatility on large indices, *Quantitative Finance,* 18, 737–748.

Lipton, Z. C. and J. Berkowitz, and C. Elkan (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning, arXivhttps://arxiv.org/abs/1506.00019

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective Approaches to Attention-based Neural Machine Translation. In *Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1412-1421).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, **86**(11), 2278-2324.

Masters, T. (1993). Practical neural network recipes in $C++$, New York: Academic Press.

Mittnik, S., Robinzonov, N., & Spindler, M. (2015). Stock market volatility: Identifying major drivers and the nature of their impact. *Journal of Banking & Finance*, 58, 1–14.

Motegi, K. C. Xiaojing, Hamori, S. and X. Haifeng (2020). Moving average threshold heterogeneous autoregressive (MAT-HAR) models, *Journal of Forecasting* 39:1035-1042.

Patton, Andrew J., and Kevin Sheppard (2009). Evaluating Volatility and Correlation Forecasts. In T. Mikosch, J.-P. KreiB, R. A. Davis, and T. G. Andersen (eds.), *Handbook of Financial Time Series*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 801–838.

Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies, *Journal of Econometrics,* 160, 246–256.

Patton, Andrew J. and Kevin Sheppard (2015). Good Volatility, Bad Volatility: Signed Jumps and the Persistence of Volatility, *Review of Economics and Statistics* 97: 683–697.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, **323**(6088), 533-536.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In NIPS 2014, arXiv:1409.3215.

Takahashi, M., Omori, Y., & Watanabe, T. (2023). *Stochastic Volatility and Realized Stochastic Volatility Models,* SpringerBriefs in Statistics. Springer Nature.

Taylor, S. J. (1982). Financial Returns Modeled by the Product of Two Stochastic Processes - a Study of the Daily Sugar Prices 1961-75, In O. D. Anderson (ed.), Time Series Analysis: Theory and Practice. Amsterdam: North-Holland, 203-226.

Terasvirta, T. (1994). Specification, Estimation, and Evaluation of Smooth Transition Autoregressive Models, *Journal of the American Statistical Association,* 89, 208-218.

Quandt, R. E. (1972). A new approach to estimating switching regressions, *Journal of the American Statistical Association* 67: 306–310. https://doi.org/10.1080/01621459.1972.10482378.

Rahimika, E. and S-H Poon (2024). Machine Learning for Realised Volatility Forecasting, Working Paper, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3707796&download=yes.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A Simple Way to Prevent Neural Networks from Over
tting. *The Journal of Machine Learning Research* 15 (1), 1929-1958.

Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., and Hinton, G. (2014). Grammar as a foreign language. Technical report, arXiv:1412.7449.

Vortelinos, D. I. (2017). Forecasting Realized Volatility: HAR against Principal Components Combining, Neural Networks and GARCH, *Research in International Business and Finance* 39, 824–839.

Wang J, Ma F, Liang C, Chen Z. (2022). Volatility forecasting revisited using Markov-switching with time-varying probability transition, *Int J Fin Econ*.27:1387–1400. https://doi.org/10.1002/ijfe.2221

Wong, Z. Y, Chin, W. C, Tan, S. H. (2016). Daily value-at-risk modeling and forecast evaluation: The realized volatility approach. *J. Finance Data Sci.* 2 (3), 171–187.

Zhang, Peter G. (2003). Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model *Neurocomputing* 50: 159–175.

Xu, K., Ba, J. L., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visualattention. In ICML'2015, arXiv:1502.03044.

Zhang, C., Zhang, Y., Cucuringu, M., & Qian, Z. (2023). Volatility Forecasting with Machine Learning and Intraday Commonality. *Journal of Financial Econometrics,* 1–39.

Zhu, H., L. Bai, L. He, and Z. Liu (2023). Forecasting realized volatility with machine learning: Panel data perspective. *Journal of Empirical Finance* 73, 251-271.

Zou, H., and T. Hastie (2005). Regularization and Variable Selection via the Elastic Net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67: 301–320.

Appendices

A Training, regularization, model evaluation, and testing

A.1 Training, validation, parameter tuning and regularization

Our training, validation, and testing approach is relatively straightforward and involves splitting data into two sets one for training and validation and another for out-of-sample testing. We

start with the initial training and validation sample, call it training sample for brevity, covering the first ten years of data for the period 01/02/1996 and 12/31/2005 and the initial test period of 01/02/2006-12/31/2006. After the initial split of the data, we extend the training window by one year and move out-of-sample test period to next year. The second training period becomes 01/02/1996-12/31/2006 with out-of-sample test period of 01/02/2007-12/31/2007. We follow this expanding window approach until all data are exhausted with the last training period ending on 12/31/2022 and out-of-sample testing period on 12/31/2023. Essentially this means that both linear and nonlinear econometric models and ML models are trained and validated on an annual basis as new information arrives with the out-of-sample forecasting analysis conducted in the following year.

We use the training and validation sample as the training set for econometric models as there is no need for cross-validation and/or hyper-parameter tuning. We tune the hyper-parameters of ML models using a five-fold time series cross validation with a grid search over a set of hyper-parameter values. The procedure begins by dividing the training/validation sample into five contiguous train and test folds or splits where each successive training set is a superset of the previous one (i.e., each training set grows in size by adding previous folds with each split), with the validation set being the next slice of the training set (i.e., the section immediately following the training set).[11] For a given training/validation and test period, each ML model is trained five times for a given set of hyper-parameter values and tested in the following validation slice. For each iteration, the model's predictive performance is evaluated using the mean squared prediction error on the validation split.

We conduct this procedure over a grid of hyper-parameter values for each model and in each training/validation period. This implies that if say there are k-different hyper-parameter combinations for a model, then the model is trained for each k-different combinations of hyper-parameter values over five splits and the specific hyper-parameter combination that gives the lowest average squared prediction errors on a validation set is selected to be the 'optimal' set of hyper-parameters and hence, the model. The model with the selected hyper-parameter values is then re-trained in the entire training sample and used in calculating forecasts in the out-of-sample test period. This procedure ensures that the model's performance is evaluated in a realistic forward-looking manner while retaining the dependence structure of the time series data. Importantly, the procedure helps identify the specific value of hyper-parameters that yield the best average performance across all training/test splits, resulting in a more robust model for forecasting, new, unseen time series data. We repeat this procedure for each training and out-of-sample test period since 2005 by expanding the training period one year and moving the out-of-sample test period next year forward.

[11] We use model selection library from Sklearn and utilize *TimeSeriesSplit* and *GridSearch* algorithms with number of splits equals to 5.

Since a standard and accepted method for determining the number of layers and number of neurons within each layer does not exist, we use a data-driven approach to determine the optimal architecture of ML models we use in this paper. Specifically, in determining the optimal architectures for DNN and RNN models in terms of the number of hidden layers and the number of units within each layer as well as some of the key hyper-parameter values, we follow the grid search with a 5-fold time series cross-validation approach discussed. This approach allows us to consider a range of network architectures ranging from a shallow network with one layer and a relatively large number neurons to a deep network with up to five layers with number of neurons in successive layers is reduced geometrically by a fixed factor of 0.5. This approach is similar to the the geometric pyramid rule suggested by Masters (1993). In other words, in addition to some of key hyper-parameters such as learning rate, drop-out rate, or the number of epochs, we let the data and our training and cross-validation approach to determine the main architecture of the model in terms of number of hidden layers and neurons within each layer over each training sample. The grid search essentially allows to consider the number of layers and units within each layer as hyper-parameters and search for the best architecture across a reasonable large set.[12]

In training DNN and RNN models, we use the ADAM optimization algorithm with Mean Squared Error as the objective function. ADAM is an adaptive learning rate optimization algorithm based on stochastic gradient descent (Kingma et al. , 2014). Other hyper-parameters tuned via the five fold time-series cross validation over the following grid values: $learning\,rate \in \{0.1,\, 0.001,\, 0.0001\}$, $batch\,size \in \{128, 512\}$, $epochs \in \{100, 250\}$, and $dropout\,rate \in \{0, 0.5, 0.8\}$. In the case of XGBoost, we tuned the following hyper-parameters: $trees \in \{5, 7, 10, 50, 100, 200\}$, $depth \in \{3, 5, 7\}$, $learning\,rate \in \{0.1, 0.01, 0.001\}$, and subsample $fraction \in \{0.5, 0.9, 1.0\}$.

When the predictor set includes additional variables, we follow the same cross-validation and hyper-parameter tuning approach above with the exception that we first identify set of relevant predictors for each training period. To this end, we select the variables through Elastic Net (EN) regularization (Zou and Hastie , 2005) by using HAR model with set of additional predictors as surrogate. The EN uses a penalized loss function,

$$\tilde{\ell}(\beta_0, \beta) = argmin_{\beta_0, \beta} \left(\sum_t \left(RV_t - \beta_0 - \beta^{'} X_{t-1} \right)^2 + \lambda \left(\alpha \sum_{i=1}^{K} \beta_i^2 + (1 - \alpha \sum_{i=1}^{K} |\beta_i| \right) \right) \quad (21)$$

where the first part of the penalized loss function, $\sum_t \left(RV_t - \beta_0 - \beta^{'} X_{t-1} \right)^2$ is the usual Least Squares loss function and the penalized part is the second expression with hyper-parameters $\lambda \geq 0$ and $\alpha \in [0,\, 1]$ which are determined via cross-validation and a grid search. The EN

[12] Specifically, the search is done over the layer/neuron set: $\{l, \{u\}\} = \{\{1, \{32\}\}, \{2, \{32, 16\}\}, \{3, \{32, 18, 8\}\}, \{4, \{32, 16, 8, 4\}\}, \{5, \{32, 16, 8, 4, 2\}\}\}$ where l is the number of layers and u is the number of neurons for each hidden layer.

regularization performs variable selection based on the penalty coefficients λ and α and hence, combines $L1$, Least Absolute (LA) and $L2$, Ridge, regularization. LA helps with variable selection by shrinking some coefficients to zero while Ridge helps handling multicollinearity by distributing the coefficient values among predictors. By combining LA and Ridge, EN allows controlling the balance between two types of regularization: as higher λ will lead more sparse model while higher α to smaller coefficient values without necessarily making them zero. In this paper, instead of selecting a specific set of values a prior for λ and α, we use again a grid search over these hyper-parameters with a five-fold time series cross validation in the training and validation sample in selecting the optimal λ and α values for each training period.[13] This allows us to determine optimal regularization parameters and hence, the set of predictors for each training period. This approach, despite considerably different optimal λ and α values generally selected all predictors except for $NEWS$ variable for majority of the training/validation samples. For three training/validation sample periods (ending 2016, 2017, and 2018), the approach eliminated size premium factor, SMB and for the first sample period ending in 2005, it removed short term interest rate change, Δi_t.

Although choosing the set of predictors across several training/validation sample through this surrogate method may not be relevant for ARFIMA, THAR, STHAR, and MSHAR as well as ML models, our experimentation with different sets of predictors resulted in very similar model performance outcomes. Moreover, since our objective is to assess performances of linear and nonlinear models relative to ML models, using the same set of predictors for a given training/validation sample period across the same set of models over time may alleviate the drawback of this approach. Additionally, to decrease the amount of over-fitting, we also use dropout which essentially works by removing some randomly selected units and their incoming and outgoing connections during the training process for DNN and RNNs (see, Srivastava etal. , 2014).[14]

A.2 Testing: Model evaluation approaches

Following the recent literature (see,for example Bucci , 2020; Christensen et al. , 2023; Zhang etal. , 2023; Branco et al. , 2024; Rahimika and Poon , 2024), we asses the performance of models by using statistical accuracy tests, Value at Risk (VaR) forecasting, and realized utility benefits. In terms of statistical accuracy and related tests, we use three approaches. First to assess each model's relative predictive performance in the out-of-sample forecasts, following Patton (2011), we compute mean squared prediction error (MSPE) and quasi-likelihood (QLIKE) for each out-of-sample test period since 2006 and for the whole test period between 2006 and 2023. Patton

[13] We used grid of values of $\lambda \in \{0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10\}$ and $\alpha = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Note that EN nests LA for $\alpha = 0$, and Ridge for $\alpha = 1$.

[14] Note also that in both scenarios, predictor variables are standardized by subtracting the sample average and dividing by the sample standard error under the ML models across each training and test samples.

(2011) indicates that the MSE and QLIKE loss functions are among the family of robust and homogeneous loss functions for volatility forecasting comparison and are robust to noisy volatility proxy. Patton and Sheppard (2009) also demonstrate that QLIKE has the highest power in the Diebold-Mariano (DM) test. We focus on both of these measures in order to assess performance of models under more than one measure. These loss measures are given by

$$MSPE = \frac{1}{T_{test}} \sum_{t=1}^{T_{test}} \left(RV_t - \widehat{RV_t} \right)^2 \tag{22}$$

$$QLIKE = \frac{1}{T_{test}} \sum_{t=1}^{T_{test}} \left[\frac{\exp(RV_t)}{\exp(\widehat{RV_t})} - \left(RV_t - \widehat{RV_t} \right) - 1 \right] \tag{23}$$

where $\widehat{RV_t}$ represents the one-day ahead predicted value of RV_t, T_{test} is the number of days in a given test period. Note that equations (22) and (23) are average values of squared prediction error and QLIKE in a out-of-sample period.

Second, we use model confidence set (MCS) proposed by Hansen et al. (2011) to identify a subset of models, $\mathcal{M}^*$ with significantly superior performance from all eleven model candidates $\mathcal{M}_0$ at a given level of confidence. The MCS procedure us based on iterative elimination of models by sequentially testing

$$H_0 : E(d_{ij,t}) = \ell_{i,t} - \ell_{j,t} = 0 \, \text{for all} \, i, j \in \mathcal{M}^*, \tag{24}$$

where $d_{ij,t}$ is the loss differential between models i and j at day t in terms of a specific loss function, such as $MSPE$ and $QLIKE$. We use T_R statistic with stationary block bootstrap of 5000 re-samples in computing the test statistic and the associated p-values. For details of the procedure, see Hansen et al. (2011). The MCS procedure allows us to make statements about the statistical significance from multiple pairwise comparisons across each test period since 2006.

Third approach we use in assessing the statistical accuracy and relative performance of models, we conduct pair-wise DM tests both in the full test period between 2006 and 2023 and each test year since 2006 by using both MSPE and QLIKE measures above. The DM test (Diebold and Marioano , 1995) evaluates the null hypothesis of equal forecasting accuracy between two models of interest and can be useful especially in cases where MCS procedure is not informative about the superior set of models.

One approach that is used in some of the recent papers in order to evaluate the economic benefit from a volatility forecasting model is to compute realized utility under some assumptions as proposed by Bollerslev et al. (2018). Since the approach is widely discussed in the recent papers (see, for example Branco et al. , 2024; Díaz et al. , 2024, for excellent descriptions of the

approahc), we do not elaborate the details of the approach for brevity. Following the literature and assuming a conditional Sharpe ratio $\gamma = 0.40$ and a coefficient of relative risk aversion $= 2$ (which implies the investor targets an annualized volatility of 20%) and assuming she uses a volatility model to define her risky position, the realized utility of using a given model denoted by $\overline{RU}^{model}$ is given by

$$\overline{RU}^{model} = \frac{1}{T_{test}} \sum_{t=1}^{T_{test}} \left(8\% \frac{\sqrt{RV_t}}{\sqrt{\widehat{RV}_t}} - 4\% \frac{RV_t}{\widehat{RV}_t} \right), \tag{25}$$

where T_{test} is the number of days in a test period under the assumption that the investor re-balances her portfolio at the end of each day. As shown by Bollerslev et al. (2018) a perfect predictive model for RV delivers a realized utility of 4% and estimated realized utility values below 4% would indicate less accurate forecasting by the model.

Given some studies argue that the RV may be useful in forecasting daily VaR (see, for example, Giot and Laurent , 2014; Wong et al. , 2016) and the importance of VaR for market participants and regulators, we build 1-day ahead VaR forecast based on the RV forecasts from different models. We use filtered historical simulation (FHS) approach (see, Barone-Adesi et al. , 1999; Alexander , 2009). To estimate VaR for day $t+1$ in a test period, we first normalize returns by dividing sample standard deviation by using the sample up to day t and the re-scale these normalized returns by the forecasted RV for day $t+1$ in the test period. Then we calculate the α-percentile of the re-scaled returns as the estimate for VaR for $t+1$ in the test period. We expand the window by one day and calculate the VaR for day $t+2$ until we exhaust all the days in a test period for each model.

More specifically, VaR estimates for the first day in the test period 01/02/2006-12/31/2006 is calculated as follows: Calculate normalized returns by dividing returns in the historical window between 01/02/1996-12/31/2005 by dividing the sample standard deviation of returns for the above period. This gives roughly 2500 days of normalized returns. Then re-scale each of these returns by multiplying with the predicted RV from a given model for the first day (i.e., 01/02/2006) and use the distribution of re-scaled returns in calculating α percentile which gives an estimate of VaR at the $1-\alpha$ confidence level. For the second day, normalize returns for the period 01/02/1996-01/02/2006 by dividing the sample standard deviation of returns in this 1-day expanded historical window size and then re-scale them by multiplying with the forecast of RV for the second day (i.e., 01/03/2006) from the model under consideration and use this distribution to compute the α-percentile. This will give an estimate of VaR for day $t+2$ in the

sample period. Continue this for each model in each out-of-sample test period.[15] We calculate VaR at $1-\alpha = 0.99$ and $1-\alpha = 0.95$ levels.

To evaluate the performance of each model in terms of VaR predictive ability, we compute the percentage of failures (exceptions) by comparing the estimated VaR for day $t+1$ (i.e., $VaR_{t+1}(\alpha)$ with the realized return r_{t+1} for each day in each of the yearly out-of-the-sample test periods since 2006 and in the entire test period between 2003 and 2023. We use likelihood ratio tests of Kupiec (1995) and Christoffersen (1998) to asses the unconditional coverage and independence property of VaR forecasts. The unconditional coverage test or the Kupiec (1995) test inspects whether the empirical failure rates are statistically equal to the theoretical value of α on average by checking if the exceptions occur as expected under the assumption that the model is correct. The second test checks if the exceptions are both independent over time and occur at the expected rate. If the model passes this conditional coverage test, it implies that both the frequency and the timing of the exceptions are on average correct.

We evaluate the VaR forecasts using the asymmetric loss function proposed by Gonzalez-Rivera et al. (2004),

$$\ell_{VaR} = \frac{1}{T_{test}} \sum_{t=1}^{T_{test}} (\alpha - d_{t+1}) \left(r_{t+1} - \widehat{VaR}^{\alpha}_{t+1} \right), \tag{26}$$

where $d_{t+1} = 1_{r_{t+1} < \widehat{VaR}^{\alpha}_{t+1}}$ is the ''hit" function. This loss function assigns a weight of $1-\alpha$ to observations for which the daily log-return is below the VaR forecast and a weight of α for returns above the VaR forecast and hence, penalizes observations for which the daily log return falls below the forecasted VaR more heavily. In order to assess the statistical accuracy of VaR forecasts, we also use DM tests and compare relative performance of models in predicting VaR under the loss function in Equation (26). See, Christensen et al. (2023) and Audrino etal. (2020) for a similar implementations.

B Results with additional predictors

Given generally the stronger performance results recorded by THAR and especially STHAR relative to other econometric models and specifically the ML set of models considered under the scenario where the predictor set included only the HAR variables, to what degree such results continue to hold under a scenario where additional predictors are included is an important question as ML models tend to pioneer in data rich environments. We repeat our empirical as-

[15] We also considered a fixed window sizes of 5 and 10 years with window size moving forward one-day as new information arrives. Since results were found to be qualitatively similar, we report and discuss results from the expanding historical window size in the paper in the following sections.

sessment of models under this second scenario by using the set of additional predictors reported in Table 8.

B.1 Set of predictors

In addition to HAR variables, this set includes features that may contain information about the state of the markets, such as the negative and positive components of 1-day, 5-day, and 22-day moving average of daily log index return, $r_t^{-/+}$. To capture the monetary policy conditions, we use daily changes in U.S. Treasury securities at 3-month constant maturity. To measure overall market appetite for risk, we use log VIX ($\log(VIX_t)$ and use log of Economic Policy Uncertainty index of (Baker etal. , 2016) ($\log(EPU_t)$). We also use a measure of news sentiment index ($NEWS_t$) due to Buckman etal. (2020) and a measure of general business conditions index due to Aruba etal. (2009). Our predictor set also includes three market factors as suggested by Fama and French (1993), including the U.S. market excess return (Rm_Rf_t), size premium factor, (i.e., average return on small stocks minus big stocks, SMB_t), and value factor (i.e, average return on value stocks minus average returns on growth stocks, HML_t). We also include relative change in RV to ensure that all models including linear HAR, ARFIMA, Markov Switching HAR (MSHAR) and ML models leverage similar information as THAR and STHAR which use the relative change in RV as threshold/transition variable.[16]

B.2 Predicting realized volatility

Table 9 reports MSPE and QLIKE in panels A and B, respectively. Results in the last rows of Panel A and B for the entire test period indicate that the top five models include HAR and its nonlinear extensions under MSPE and HAR, THAR, STHAR, and DNN and BRNN under QLIKE. MCS procedure includes all models in the confidence set under QLIKE includes all but ARFIMA under MSPE in the entire full test sample. Despite slight differences, these results are broadly in line with the results reported in Table 2.

Comparison of MSPE and QLIKE results between Tables 2 and 9 reveals that consistent with the expanded information with the additional predictors, all models achieve smaller prediction errors both in the full test period and over majority of the separate test periods. While the prediction error gains are generally across the board for all models, HAR, THAR, and STHAR models tend to achieve lower MSPE and QLIKE compared to other models and especially ML models. Overall, differences in prediction errors among models become smaller especially under QLIKE as the careful inspection of Tables 2 and 9 would reveal. It worth also noting that the difference between STHAR and HAR and THAR model becomes smaller and we note more test periods with similar error performance among these three models. This is likely driven by the

[16] We lag all predictor variables apart from the HAR variables by one day.

Table 8: List of predictors used

Variable	Description	Source
$RV_{d,t}$	1-day moving average of RV	Bloomberg
$RV_{w,t}$	5-day moving average of RV	Bloomberg
$RV_{m,t}$	22-day moving average of RV	Bloomberg
$\Delta RV_t/RV_{t-1}$	Daily relative change in RV	Bloomberg
$r_{d,t}^{-/+}$	1-day moving average of negative/positive index return	Bloomberg
$r_{w,t}^{-/+}$	5-day moving average of negative/positive index return	Bloomberg
$r_{m,t}^{-/+}$	22-day moving average of negative/positive index return	Bloomberg
Δi_t	Daily change in U.S. 3M Treasury	FRED
$\log(VIX_t)$	Log VIX index	FRED
ADS_t	Business Conditions Index (Aruba etal., 2009)	FRB Philadelphia
$\log(EPU_t)$	Log Economic Policy Uncertainty Index (Baker etal., 2016)	FRED
$NEWS_t$	News sentiment index (Buckman etal., 2020)	FRB San Francisco
Rm_Rf_t	US market excess return	K. French's webpage
SMB_t	Size premium factor	K. French's webpage
HML_t	Value factor	K. French's webpage

The underlying daily RV estimates are constructed using 5-minute intraday data on open and close prices of S&P 500 index which are obtained from Bloomberg. S&P 500 index returns are based on the daily open and close prices from Bloomberg. Data on the ADS index, news sentiment index, EPU, and Fama-French factors are available at https://www.philadelphiafed.org/surveys-and-data/real-time-data-research/ads, https://www.frbsf.org/research-and-insights/data-and-indicators/daily-news-sentiment-index/, https://fred.stlouisfed.org/categories/33201, and https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/Data_Library/f-f_factors.html, respectively. For each variable the source indicates the source of the undelrying rwa data. All transformations are author's own calculation.

fact THAR and specifically STHAR model is over parameterized compared to the HAR model under extended predictor set and hence, the difference in the prediction error as measured by MSPE and QLIKE diminishes in many of the test periods.

Above observations in terms of reduced differences in prediction errors are reflected themselves in the MCS procedure results as now the procedure includes more models in the confidence set for majority of test periods. Nevertheless, despite being over-parameterized, THAR and STHAR models continue perform well and compete or outperform ML models in several test periods where MCS procedure leads to a narrow set of models in the confidence set. Potentially as a function of improvements in the prediction errors, the MCS procedure includes all models in the MCS in 2008, 2010, 2011, 2013, 2015, and 2020. In the remaining test years, the MCS procedure eliminates generally ARFIMA and occasionally one or two of the ML models.[17]

Tables 10 and 11 present pairwise DM test results for selected test periods and the entire test sample in the last panels. Inspection of the results show that HAR and THAR attain statistically significant lower MSPE in the full test period sample against all models except for STHAR. HAR and THAR outperform all but STHAR by achieving statistically significantly

[17]Such expanded MCS might be also due to the fact that all models now have more parameters to estimate and learn and hence, the data might have become less informative due to extended prediction set compared to the scenario where only past values of RV are included in the features set. Further exploration of this phenomenon in the context of nonlinear and ML models are left for future work.

Table 9: One-day ahead MSPE and QLIKE over test periods with additional predictors

YEAR	HAR	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTAM-A
Panel A. One-day ahead MSPEs											
2006	0.011	0.011	0.012	0.010	0.015	0.011	0.012	0.011	0.011	0.012	0.011
2007	0.039	0.040	0.044	0.040	0.047	0.042	0.040	0.041	0.040	0.042	0.045
2008	0.290	0.316	0.378	0.280	0.399	0.626	0.399	0.432	0.458	0.528	0.460
2009	0.041	0.046	0.056	0.040	0.057	0.045	0.044	0.044	0.049	0.044	0.043
2010	0.057	0.053	0.059	0.058	0.066	0.053	0.059	0.059	0.058	0.058	0.059
2011	0.057	0.051	0.064	0.058	0.064	0.058	0.057	0.056	0.056	0.059	0.053
2012	0.017	0.017	0.018	0.017	0.025	0.016	0.018	0.019	0.019	0.019	0.018
2013	0.018	0.019	0.018	0.016	0.024	0.018	0.018	0.017	0.018	0.017	0.017
2014	0.017	0.015	0.017	0.016	0.020	0.016	0.020	0.020	0.018	0.019	0.017
2015	0.060	0.056	0.066	0.060	0.069	0.051	0.061	0.058	0.062	0.058	0.060
2016	0.022	0.021	0.024	0.022	0.033	0.019	0.022	0.022	0.022	0.022	0.027
2017	0.007	0.007	0.007	0.007	0.012	0.008	0.007	0.010	0.009	0.009	0.007
2018	0.035	0.034	0.042	0.039	0.040	0.043	0.038	0.039	0.038	0.040	0.047
2019	0.019	0.020	0.020	0.020	0.026	0.019	0.020	0.020	0.021	0.021	0.021
2020	0.126	0.147	0.159	0.114	0.179	0.243	0.156	0.137	0.132	0.133	0.133
2021	0.032	0.031	0.033	0.030	0.037	0.031	0.034	0.033	0.032	0.032	0.034
2022	0.064	0.066	0.073	0.066	0.080	0.063	0.069	0.069	0.069	0.068	0.067
2023	0.021	0.022	0.023	0.020	0.028	0.022	0.022	0.022	0.022	0.022	0.023
06-23	0.052	0.054	0.062	0.059	0.069	0.078	0.062	0.063	0.064	0.068	0.064
Panel B. One-day ahead QLIKE											
2006	0.005	0.006	0.006	0.005	0.008	0.006	0.006	0.005	0.006	0.006	0.006
2007	0.021	0.022	0.026	0.022	0.025	0.023	0.022	0.022	0.022	0.023	0.024
2008	0.287	0.434	0.655	0.280	0.785	1.836	0.658	0.825	0.944	1.265	0.985
2009	0.022	0.024	0.033	0.021	0.031	0.024	0.024	0.023	0.028	0.024	0.022
2010	0.040	0.037	0.052	0.040	0.056	0.040	0.045	0.043	0.047	0.048	0.047
2011	0.031	0.028	0.044	0.032	0.038	0.030	0.034	0.033	0.033	0.039	0.029
2012	0.009	0.008	0.009	0.008	0.013	0.008	0.009	0.010	0.010	0.010	0.009
2013	0.009	0.010	0.010	0.009	0.013	0.009	0.009	0.009	0.009	0.009	0.009
2014	0.008	0.007	0.009	0.008	0.011	0.008	0.010	0.010	0.009	0.010	0.009
2015	0.069	0.061	0.101	0.067	0.097	0.053	0.078	0.068	0.083	0.071	0.072
2016	0.011	0.011	0.014	0.012	0.018	0.010	0.012	0.012	0.012	0.012	0.014
2017	0.003	0.004	0.004	0.004	0.006	0.004	0.004	0.005	0.005	0.005	0.004
2018	0.019	0.019	0.026	0.022	0.022	0.022	0.021	0.022	0.021	0.022	0.029
2019	0.010	0.010	0.011	0.010	0.013	0.010	0.011	0.011	0.011	0.011	0.011
2020	0.075	0.117	0.123	0.065	0.143	0.133	0.087	0.090	0.079	0.081	0.077
2021	0.016	0.016	0.018	0.015	0.019	0.015	0.017	0.017	0.017	0.017	0.018
2022	0.035	0.035	0.044	0.035	0.046	0.034	0.039	0.038	0.041	0.040	0.038
2023	0.011	0.011	0.013	0.011	0.015	0.011	0.011	0.012	0.011	0.012	0.012
06-23	0.038	0.048	0.067	0.056	0.077	0.078	0.062	0.063	0.064	0.068	0.064

Notes: The table reports the 1-day ahead out-of-sample Mean Squared Prediction Error (MSPE) and Quasi-Likelihood (QLIKE)loss for each model for each of the test samples over the period 2006 and 2023. The last row reports MSPE and QLIKE for the entire test period of 2006-2023. Values in blue indicate models that are included in the MCS at the 5% significance level. Years in which all models are in MCS are indicated in red. Values in black indicate models that are not in the MCS.

lower MSPE in the full sample while STHAR beats ARFIMA, XGB, and LSTM models under MSPE. Under QLIKE, results are less pronounced despite HAR, THAR, and STHAR models continue to achieve lower QLIKE loss relative to most models with statistically lower QLIKE against some of the ML models including DNN, BRNN, GRU, and LSTM-A models at %5 significance level. Results for the full test period in the last row of Table 11 are in-line with the MCS procedure as no model statistically dominates in terms of attaining statistically lower QLIKE.

Reported pair-wise DM test results for 2008 in Panel A of Table 10 show HAR and THAR display superior performance by achieving statistically lower MSPE against all but STHAR model which statistically outperforms all ML models except for DNN. DM test results in Panel A of Table 11 show no statistically meaningful difference in average QLIKE between most model pairs with especially the exception of the statistically lower QLIKE attained by STHAR against BRNN, GRU, and LSTM-A. We also note that THAR attains statistically significant lower QLIKE than DNN. In 2020, despite HAR and THAR models attain lower MSPE and QLIKE, DM tests distinguish their better performance only against MSHAR, ARFIMA, XGB, and DNN both in terms of MSPE and QLIKE differences as can be seen from Panel B of each table. In 2022, HAR generally obtains statistically lower MSPE and QLIKE than most models with majority of the models not performing better than each other. Similar results generally follows for 2023 where prediction error differences diminish among models. These observations suggest that as new data covering periods of elevated volatility introduced in the training and validation of models, the difference in performance between linear HAR and its nonlinear versions, THAR and STHAR and ML models tend to diminish, leading no statistically different prediction error differentials.

Table 10: DM test under MSPE with extended predictors over selected test periods and the entire test period

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
A. Test Period 2008										
HAR	-0.94	-1.99*	-0.53	-1.98*	-3.02*	-2.47*	-2.52*	-2.63*	-2.94*	-2.49*
THAR		-1.80	-0.09	-2.16*	-3.28*	-2.37*	-2.61*	-2.81*	-3.22*	-2.67*
MSHAR			1.18	-0.83	-3.37*	-2.50*	-3.62*	-3.67*	-3.88*	-3.04*
STHAR				-1.75	-3.45*	-1.57	-2.12*	-2.47*	-3.13*	-2.38*
FI					-3.32*	0.05	-1.16	-1.97*	-3.15*	-1.96
XGB						3.02*	3.03*	2.91*	2.22*	3.18*
DNN							-2.35*	-2.73*	-3.31*	-2.15*
BRNN								-3.32*	-3.75*	-1.55
GRU									-3.93*	-0.13
LSTM										3.46*
B. Test Period 2020										

Table 10 continued from previous page

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
HAR	-1.20	-1.98*	-1.03	-2.36*	-2.70*	-3.27*	-1.46	-1.19	-1.22	-1.44
THAR		-0.76	0.06	-2.02*	-2.31*	-0.39	0.77	0.98	0.94	0.85
MSHAR			0.51	-1.79	-1.94	0.16	1.94	1.68	1.67	1.47
STHAR				-1.12	-1.94	-0.37	0.45	0.68	0.62	0.64
FI					-1.46	0.98	2.62*	2.32*	2.36*	2.12*
XGB						2.15*	2.53*	2.69*	2.67*	2.64*
DNN							1.87	3.32*	2.99*	3.34*
BRNN								0.95	0.81	0.58
GRU									-1.07	-0.58
LSTM										0.07
C. Test Period 2022										
HAR	-0.94	-1.75	-2.00*	-3.08*	0.46	-2.32*	-3.16*	-1.28	-1.19	-1.17
THAR		-1.04	-1.56	-2.02*	1.07	-0.65	-0.87	-0.47	-0.37	-0.09
MSHAR			-0.56	-1.72	1.64	1.11	0.85	1.82	1.97*	1.74
STHAR				-0.21	2.02*	1.28	1.21	1.16	1.26	1.54
FI					2.81*	2.60*	2.30*	2.90*	3.14*	3.21*
XGB						-1.75	-2.04*	-1.23	-1.16	-1.08
DNN							-0.52	-0.02	0.28	2.03*
BRNN								0.13	0.38	1.83
GRU									1.80	1.16
LSTM										0.96
D. Test Period 2023										
HAR	-1.34	-1.35	-2.25*	-4.29*	-1.14	-1.72	-2.30*	-1.64	-2.23*	-1.91
THAR		-0.53	-1.57	-2.70*	0.35	0.18	-0.35	0.23	-0.30	-0.70
MSHAR			-1.24	-2.35*	0.89	0.53	0.23	0.59	0.26	-0.14
STHAR				-0.86	1.81	1.56	1.30	1.65	1.32	1.27
FI					3.38*	3.83*	3.72*	3.78*	3.84*	2.55*
XGB						-0.24	-0.69	-0.20	-0.66	-1.18
DNN							-1.95	0.12	-1.68	-1.19
BRNN								1.21	0.62	-0.61
GRU									-1.05	-1.25
LSTM										-0.67
E. Full Test Period: 2006-2023										
HAR	-1.10	-3.47*	-1.69	-4.51*	-3.69*	-3.52*	-3.07*	-3.11*	-3.31*	-3.02*
THAR		-3.24*	-1.19	-5.21*	-3.90*	-2.93*	-2.88*	-2.97*	-3.37*	-2.91*
MSHAR			1.09	-3.35*	-3.04*	0.62	0.05	-0.85	-2.17*	-0.86
STHAR				-2.85*	-3.21*	-0.87	-1.10	-1.44	-2.21*	-1.48
FI					-1.94	3.18*	3.08*	2.17*	0.24	1.88
XGB						3.28*	3.47*	3.34*	2.79*	3.47*
DNN							-0.80	-1.75	-2.75*	-1.55

Table 10 continued from previous page

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
BRNN								-2.21*	-3.55*	-1.58
GRU									-3.66*	-0.35
LSTM										2.93*

Notes: The table reports Diebold-Mariano test for the equality of MSPE between the model on the row and the model on the column over selected test periods under the scenario where predictor set includes additional variables A * indicates rejection of the null against the one-sided alternative under the 5% significance. A positive value indicates that the MSPE of the model on the row is greater than the model on the column.

Table 11: Diebold-Mariano under QLIKE with extended predictors over selected test periods and the entire test sample

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
Panel A. Test Period 2008										
HAR	-1.09	-1.49	-0.94	-1.26	-1.43	-1.61	-1.47	-1.45	-1.42	-1.46
THAR		-1.87	-0.79	-1.34	-1.48	-2.22*	-1.68	-1.59	-1.50	-1.59
MSHAR			0.92	-0.84	-1.41	-0.02	-1.43	-1.39	-1.38	-1.42
STHAR				-1.83	-1.61	-0.82	-2.51*	-2.16*	-1.75	-2.07*
ARFIMA					-1.53	0.75	-0.76	-2.35*	-1.63	-2.26*
XGB						1.38	1.40	1.41	1.42	1.40
DNN							-1.23	-1.27	-1.32	-1.31
BRNN								-1.33	-1.36	-1.39
GRU									-1.37	-1.20
LSTM										1.32
Panel B. Test Period 2020										
HAR	-1.51	-2.39*	-0.78	-1.98*	-2.25*	-1.88	-1.64	-1.02	-1.16	-0.60
THAR		-0.50	1.15	-2.05*	-0.94	1.22	1.39	1.55	1.55	1.54
MSHAR			1.84	-1.21	-0.55	2.01*	2.75*	2.53*	2.58*	2.42*
STHAR				-1.73	-1.78	-0.30	-0.53	0.32	0.15	0.59
ARFIMA					0.61	1.83	2.09*	2.07*	2.08*	2.03*
XGB						2.14*	2.34*	2.39*	2.41*	2.34*
DNN							-0.49	1.74	1.20	2.10*
BRNN								1.87	1.88	1.74
GRU									-1.50	1.27
LSTM										1.40
Panel C. Test Period 2023										
HAR	-0.26	-2.81*	-1.54	-3.09*	0.76	-3.29*	-3.85*	-2.55*	-2.45*	-2.40*
THAR		-2.24*	-1.40	-2.35*	0.71	-1.70	-1.74	-1.76	-1.63	-1.25
MSHAR			0.77	-0.57	2.71*	2.21*	2.06*	2.20*	2.49*	2.55*
STHAR				-0.99	1.73	0.35	0.45	-0.16	-0.02	0.50

Table 11 continued from previous page

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
ARFIMA					3.09*	2.35*	2.28*	2.08*	2.36*	2.75*
XGB						-2.71*	-2.78*	-2.48*	-2.38*	-2.21*
DNN							0.79	-1.50	-1.16	1.27
BRNN								-1.38	-1.11	0.41
GRU									3.28*	2.38*
LSTM										2.10*
Panel D. Test Period 2023										
HAR	-1.06	-1.82	-2.32*	-4.20*	-1.53	-1.72	-2.25*	-1.72	-2.20*	-2.26*
THAR		-1.14	-1.73	-2.68*	-0.16	-0.09	-0.54	-0.02	-0.52	-1.18
MSHAR			-0.64	-1.72	1.34	1.10	0.82	1.20	0.85	0.26
STHAR				-1.07	1.66	1.57	1.29	1.71	1.28	1.00
ARFIMA					3.28*	3.69*	3.60*	3.66*	3.70*	2.30*
XGB						0.12	-0.36	0.19	-0.35	-1.35
DNN							-1.85	0.18	-1.72	-1.71
BRNN								1.11	0.14	-1.10
GRU									-1.03	-1.72
LSTM										-1.14
Panel E. Full test sample: 2006-2023										
HAR	-1.28	-2.04*	-0.99	-1.68	-1.47	-1.79	-1.56	-1.55	-1.48	-1.52
THAR		-2.64*	-0.70	-1.86	-1.49	-2.22*	-1.68	-1.63	-1.52	-1.59
MSHAR			1.95	-1.02	-1.28	2.64*	-0.45	-0.91	-1.14	-0.92
STHAR				-2.82*	-1.64	-1.03	-2.58*	-2.28*	-1.81	-2.12*
FI					-1.33	1.48	1.65	-0.40	-1.17	-0.56
XGB						1.37	1.41	1.40	1.42	1.41
DNN							-1.15	-1.29	-1.32	-1.27
BRNN								-1.48	-1.39	-1.38
GRU									-1.34	-0.68
LSTM										1.36

Notes: The table reports Diebold-Mariano test for the equality of QLIKE between the model on the row and the model on the column over three test periods with additional predictors. A * indicates rejection of the null against the one-sided alternative under the 5% significance. A positive value indicates that the QLIKE of the model on the row is greater than the model on the column.

B.3 Realized utility with extended set of predictors

Reported results on realized utility benefits from using each model for predicting volatility in Table 12 is similar to the reported results in Table 5 as generally most models provide similar realized utility benefits in the full test period and in most periods except for periods with

Table 12: Realized Utility over test periods

Year	HAR	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
2006	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2007	0.040	0.040	0.039	0.040	0.039	0.040	0.040	0.040	0.040	0.040	0.040
2008	0.032	0.027	0.020	0.022	0.015	-0.022	0.020	0.014	0.010	-0.001	0.008
2009	0.040	0.040	0.039	0.039	0.039	0.040	0.040	0.040	0.039	0.040	0.040
2010	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2011	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2012	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2013	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2014	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2015	0.038	0.038	0.037	0.039	0.037	0.039	0.038	0.038	0.038	0.038	0.038
2016	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2017	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2018	0.040	0.040	0.039	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.039
2019	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2020	0.038	0.037	0.037	0.038	0.036	0.037	0.038	0.038	0.038	0.038	0.038
2021	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
2022	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
2023	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040	0.040
06-03	0.039	0.039	0.038	0.039	0.038	0.036	0.038	0.038	0.038	0.037	0.038

Notes: The table reports the realized utility benefit of using forecasts of volatility based on different models under the assumptions of a constant conditional Sharpe ratio equal to 0.40 and a coefficient of risk aversion $\gamma = 2$. The maximum utility benefit in this setting using the future realized volatilities, is equal to 4

elevated volatility. For example, in 2008, HAR continues to provide the highest realized utility (3.2%) compared to the relatively dismal performances of ML models. In 2020, the difference in realized utility benefit between econometric and ML models diminish, a result consistent with the improvements in the predictive errors of all models.

B.4 Predicting VaR with additional predictors

Table 13 reports the failure rates across each test period and in the entire test sample since 2006 with unconditional and conditional coverage test results. A striking difference between the results under the scenario with additional predictors is that the coverage rates for all models display a significant improvement compared to the case where only HAR variables are included in the predictor set. This marked difference can be observed by comparing the reported VaR exceedance rates and Kupiec and Christoffersen test results in Tables 6 and 13. Results in the latter table clearly demonstrate the benefits of using more information by including more variables in the estimation of linear and nonlinear econometric models as well as ML models. Kupiec test rarely rejects the null of correct unconditional coverage at 5% significance level of VaR across each test sample since 2006 for nearly all models. Only exceptions are STHAR model in 2017 and LSTM model in 2008. Christoffersen test never reject its null hypothesis, suggesting statistically correct conditional coverage by all models in each test periods since

Table 13: VaR Exceedance Rates & Kupiec and Chiristoffersen Tests over test periods and the entire test sample

Year	HAR	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
2006	0.004	0.008	0.004	0.016	0.004	0.004	0.008	0.008	0.004	0.004	0.004
2007	0.016	0.016	0.016	0.028*	0.020	0.016	0.016	0.012	0.024	0.008	0.004
2008	0.008	0.012	0.016	0.020	0.012	0.016	0.024	0.024	0.024	0.028*	0.012
2009	0.004	0.004	0.004	0.008	0.000	0.004	0.004	0.004	0.004	0.000	0.000
2010	0.008	0.004	0.016	0.012	0.016	0.008	0.012	0.012	0.016	0.016	0.016
2011	0.016	0.016	0.024	0.012	0.020	0.012	0.020	0.016	0.016	0.024	0.020
2012	0.008	0.008	0.012	0.000	0.012	0.000	0.000	0.000	0.000	0.000	0.000
2013	0.008	0.004	0.004	0.016	0.016	0.004	0.012	0.004	0.012	0.004	0.004
2014	0.004	0.004	0.008	0.016	0.012	0.000	0.004	0.004	0.008	0.004	0.008
2015	0.004	0.008	0.008	0.004	0.004	0.004	0.004	0.004	0.004	0.004	0.008
2016	0.008	0.008	0.008	0.008	0.008	0.004	0.008	0.012	0.012	0.012	0.012
2017	0.012	0.012	0.012	0.012	0.016	0.008	0.012	0.012	0.012	0.012	0.012
2018	0.008	0.008	0.012	0.000	0.012	0.004	0.012	0.012	0.012	0.012	0.020
2019	0.000	0.008	0.008	0.012	0.008	0.000	0.004	0.000	0.008	0.004	0.000
2020	0.012	0.016	0.012	0.008	0.008	0.004	0.012	0.012	0.012	0.012	0.012
2021	0.004	0.004	0.004	0.008	0.004	0.000	0.004	0.004	0.004	0.004	0.000
2022	0.004	0.004	0.008	0.012	0.008	0.004	0.004	0.004	0.016	0.016	0.016
2023	0.000	0.000	0.004	0.009	0.008	0.000	0.000	0.000	0.000	0.000	0.004
06-23	0.007*	0.008	0.010	0.011	0.010	0.005*†	0.009	0.008	0.010	0.009	0.008

Notes: The table reports VaR Exceedance Rates over each test year and in the entire test period between 2006 and 2023 across models with additional predictors. The exceedance rate is defined to be the number of days in a test period where log returns falls below the predicted VaR divided by the number of VaR days in a period. A * indicates the rejection of the null hypothesis of correct unconditional coverage (i.e., rejection of the null by the Kupiec test) at 5 percent significance level that the exceedance rate equals to 1 percent (i.e., the VaR confidence level of 99 percent) against the one-sided alternative that it is more than 1 percent. A †indicate the rejection of the correct conditional coverage rate at 5% significance level. The last row gives results for the entire test period, 2006-2023.

2006. In the full test period between 2006 and 2023, correct unconditional coverage is rejected under HAR and XGB while all other models remain to attain correct coverage rates.

Inspecting the results in Tables 6 and 13 clearly demonstrate that STHAR is the only model with nearly perfect unconditional and conditional coverage irrespective of whether we include more predictors or simply rely on past values of RV. This outcome gives a strong reason for practitioners and market participants to consider such nonlinear alternatives as they may not require the additional cost and burden of retaining data on a large number of predictors.

DM test results for selected test periods and the entire test sample are reported in Table 14. Inspection of the results in this table reveals a number of striking results especially compared to the results in Table 13 and DM test results when only the past RV are included in the prediction set. First, in the full test period reported in Panel E of the Table, MSHAR model attains statistically lower VaR loss relative to all models except for HAR and THAR. In this sense, it outperforms all ML models and ARFIMA and STHAR model in the full test sample. With the few exceptions, all remaining models perform relatively equally in terms of VaR predictive accuracy under the asymmetric VaR loss function considered.

Table 14: DM tests of predictive accuracy of VaR over selected test periods with additional predictors

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
A. Test Period 2008										
HAR	-0.91	-0.5	-1.52	-1.53	-1.22	-1.17	-1.22	-1.23	-1.34	-0.77
THAR		0.59	-1.66	0.08	-1.25	-1.11	-1.2	-1.21	-1.37	-0.55
MSHAR			-2.15*	-0.41	-1.40	-1.80	-1.76	-1.70	-1.77	-1.04
STHAR				1.48	-0.12	2.31*	2.40*	2.33*	0.34	3.62*
ARFIMA					-1.09	-1.01	-1.10	-1.12	-1.27	-0.50
XGB						0.88	0.70	0.60	0.27	1.32
DNN							-1.55	-1.44	-1.57	1.85
BRNN								-1.22	-1.42	2.40*
GRU									-1.33	2.30*
LSTM										2.23*
B. Test Year 2020										
HAR	1.25	0.81	1.63	1.02	1.08	0.48	1.33	0.77	0.81	0.87
THAR		-0.26	1.43	0.41	0.93	-0.19	0.3	-0.03	0.06	0.19
MSHAR			1.07	0.51	0.85	0.02	0.43	0.16	0.21	0.29
STHAR				-0.94	0.59	-1.78	-1.36	-1.38	-1.35	-1.42
ARFIMA					0.97	-1.15	-0.55	-0.93	-0.87	-0.62
XGB						-1.28	-1.00	-1.14	-1.14	-1.14
DNN							1.09	0.59	1.08	3.96*
BRNN								-0.83	-0.56	-0.12
GRU									1.02	0.84
LSTM										0.68
C. Test Year 2022										
HAR	0.62	5.55*	-0.77	-0.49	0.92	8.86*	5.35*	-0.94	-0.88	-0.78
THAR		6,67*	-0.95	-0.63	1.07	2.82*	2.84*	-0.94	-0.89	-0.78
MSHAR			-2.37*	-1.95	-2.26*	-3.70*	-4.36*	-1.96	-2.08*	-2.68*
STHAR				0.008	1.35	1.45	1.37	-0.24	-0.15	0.40
ARFIMA					0.96	0.98	0.93	-0.37	-0.24	0.11
XGB						0.20	0.04	-1.02	-0.99	-0.92
DNN							-1.97*	-1.37	-1.39	-1.66
BRNN								-1.29	-1.29	-1.47
GRU									1.14	0.94
LSTM										0.79
D. Test Year 2023										
HAR	0.28	2.60*	-1.34	-0.81	1.12	-0.77	0.70	-0.14	0.21	2.17*
THAR		2.53*	-1.35	-0.88	0.73	-0.81	0.23	-0.37	-0.11	1.84
MSHAR			-1.84	-3.98*	-2.37*	-2.8*	-2.30*	-2.59*	-2.45*	-1.86
STHAR				1.31	1.41	1.32	1.37	1.33	1.36	1.58
ARFIMA					1.08	0.72	0.94	0.80	0.86	2.02*
XGB						-1.44	-0.47	-1.01	-0.78	1.47

Table 14 continued from previous page

Models	THAR	MSHAR	STHAR	ARFIMA	XGB	DNN	BRNN	GRU	LSTM	LSTM-A
DNN							3,10*	1.34	2.09*	2.67*
BRNN								-1.41	-4.39*	1.84
GRU									0.56	2.44*
LSTM										2.07*
E. Test period between 2006 and 2023										
HAR	-0.89	0.6	-1.93	-3.76*	-0.15	-1.41	-1.32	-1.68	-1.78	-1.73
THAR		1.72	-1.97*	-2.72*	0.11	-1.24	-1.16	-1.63	-1.77	-1.58
MSHAR			-2.74*	-4.39*	-0.37	-2.85*	-2.56*	-2.92*	-2.9*	-2.97*
STHAR				0.19	1.94	1.66	1.56	0.9	0.37	0.72
ARFIMA					1.94	1.31	1.07	0.5	0.1	0.48
XGB						1.22	-0.87	-1.29	-1.74	-1.18
DNN							-0.82	-1.79	-2.11*	-1.19
BRNN								-0.42	-2.41*	-1.05
GRU									-1.28	-0.03
LSTM										1.73

Notes: The table reports Diebold-Mariano test of equal predictive accuracy of predicted VaR under VaR loss function for the test years 2008, 2020, 2022, and 2023. The null hypothesis being tested is $H_0 : E(\ell_{VaR_i}) = E(\ell_{VaR_j})$ against $H_0 : E(\ell_{VaR_i}) > E(\ell_{VaR_j})$, where model i is the label of the selected row, whereas model j is the label of the selected column. A * indicates rejection of the null against the one-sided alternative under the 5% significance. A positive value indicates that the average loss under the model on the row is greater than the model on the column.

Second, moving to individual test periods reveals that MSHAR display a strong performance by beating most or all of the models in 2022 and 2023 as pairwise DM test strongly supports the performances of MSHAR against all models. This strong performance by MSHAR disappear in 2008 and 2020 as volatility reaches extremely high values during these test periods and the difference between models diminish. Yet a third result is noted in Panel A of Table 14, ML models including DNN, BRNN, GRU, and LSTM-A beats STHAR model in 2008.

C Additional results for RNNs under long time steps

To ensure consistency between econometric and ML models in terms of the information set used, we presented results based on HAR predictors in the main paper and extended these with additional predictors in Appendix B. While HAR predictors, particularly $RV_{w,t}$ and $RV_{m,t}$, inherently encode some historical information, using sequences of past observations can enable RNN models to capture more complex, nonlinear dynamics over time. This raises the question of whether RNN performance improves with longer time steps, allowing these models to better capture temporal dependencies.

Table 15: One-day ahead MSPE and QLIKE over test periods with 21 day-time-steps for RNNs

	MSPE				QLIKE			
YEAR	BRNN	GRU	LSTM	LSTM-A	BRNN	GRU	LSTM	LSTM-A
2006	0.018	0.018	0.018	0.018	0.009	0.009	0.009	0.009
2007	0.073	0.074	0.071	0.068	0.040	0.039	0.038	0.038
2008	0.680	0.633	0.800	0.557	2.196	2.090	3.441	1.758
2009	0.056	0.052	0.056	0.057	0.031	0.028	0.029	0.029
2010	0.102	0.100	0.095	0.094	0.107	0.112	0.107	0.108
2011	0.104	0.099	0.096	0.103	0.062	0.061	0.062	0.065
2012	0.027	0.027	0.027	0.027	0.014	0.014	0.014	0.014
2013	0.034	0.030	0.031	0.038	0.017	0.016	0.016	0.019
2014	0.030	0.031	0.031	0.032	0.016	0.016	0.017	0.017
2015	0.120	0.121	0.114	0.115	0.293	0.298	0.295	0.301
2016	0.271	0.034	0.034	0.034	0.085	0.019	0.019	0.019
2017	0.035	0.019	0.014	0.017	0.017	0.009	0.007	0.008
2018	0.085	0.084	0.081	0.079	0.053	0.051	0.049	0.049
2019	0.037	0.038	0.038	0.037	0.020	0.021	0.021	0.020
2020	0.259	0.296	0.280	0.316	0.196	0.350	0.304	0.373
2021	0.054	0.051	0.051	0.052	0.028	0.027	0.027	0.027
2022	0.099	0.097	0.096	0.097	0.063	0.058	0.057	0.058
2023	0.029	0.030	0.031	0.030	0.016	0.016	0.016	0.016

Notes: See, Table 2 for explanations.

RNNs, including LSTM and LSTM with Attention architectures, are particularly well-suited for modeling how patterns in realized volatility evolve over time, potentially uncovering dynamics beyond those captured by moving averages. Given the persistence and clustering often observed in realized volatility, longer sequences may allow RNNs to better learn these effects, even when they span several weeks or months. To evaluate the robustness of our findings for RNNs with long time steps, we extend the time step to 22 days (roughly a month) and present the results in Table 15.

A comparison of the average MSPE and QLIKE results for RNNs in Table 15 with those in Table 2 reveals that RNNs with longer time steps yield higher average MSPE and QLIKE values. This is in contrast to results for RNNs under the baseline scenario where the predictor set includes $RV_{d,t}$, $RV_{w,t}$ and $RV_{m,t}$ across test periods since 2006. These findings suggest that the HAR predictors, particularly the weekly and monthly averages, already capture the dynamics of realized volatility sufficiently. Consequently, RNN models with time steps of 22 days do not gain additional insights into nonlinear dynamics beyond those encoded in the HAR predictors.

This is an important result, as training and validating RNNs, especially LSTM and LSTM-A models, with long time steps is considerably more complex and time-consuming. By demon-

strating that the simpler HAR predictors without time steps beyond 1-day are adequate, this finding supports the robustness of our conclusions regarding the performance of econometric and ML models, including RNNs, relative to linear HAR and its nonlinear extensions.[18]

[18]While these results are striking, further investigation is warranted to fully understand their implications for RV modeling. We leave this exploration for future research.